Some results on the Generalized Weihrauch Hierarchy

MSc Thesis *(Afstudeerscriptie)*

written by

Gian Marco Osso
(born August 7th, 1998 in Cetraro (CS), Italy)

under the supervision of **Dr. Lorenzo Galeotti** and **Prof. Dr. Benedikt Löwe**, and submitted to the Examinations Board in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam.*

Date of the public defense:	**Members of the Thesis Committee:**
August 29th, 2022	Dr. Ekaterina Shutova (Chair)
	Dr. Merlin Carl
	Dr. Lorenzo Galeotti
	Dr. Levin Hornischer
	Prof. Dr. Benedikt Löwe
	Prof. Dr. Yde Venema

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

Acknowledgements

First of all, I would like to thank my supervisors Lorenzo and Benedikt, who enabled me to write this thesis, which would not exist without their guidance. In particular, I thank Lorenzo for introducing me to the topics of Weihrauch degrees and generalized analysis, and for helping me throughout the writing process. Our often exhausting meetings would always remind me of why I love doing mathematics and they would motivate me during the harder times of my last semester. I thank Benedikt for his invaluable advice on writing style and mathematical storytelling, and for making me understand how crucial these things are in the practice of research. I can safely say that both of you had a huge impact on the quality of my thesis, as well as on me as a potential future researcher.

Thanks to the other defense committee members, who put in the time to read my not-so-short thesis and understood it to the point of being able to ask such thoughtful questions.

I thank my family who, despite having no idea of what I have been studying the past five years, continues to support me in all I do. The trust you have shown me and the support you have always provided are remarkable, and the fact that I know I can always count on you has helped me immensely in these past two years of living abroad.

I thank my friends in Italy for being willing to keep our relationships alive despite the distance: with some of you our relationship is long-distance since two years, for some of you it is already five years of being apart, but - I can say this about all of you - the bond between us has not faded a little bit. I know that this requires effort on both sides, and I am grateful for your availability. I hope we will continue to feel as connected as we have always felt in the once-a-year meetings to follow.

Lastly I thank my friends in Amsterdam for these past two years we spent together. You are the reason why I have felt at home (literally, for some of you) during my masters and you are a large part of what drove my growth in the past two years. Our passionate conversations, adventurous days and nights, the sports, long walks, music and food we shared are a big part of what I enjoyed in Amsterdam, and they have been and continue to be an inspiration for me. Thanks for the great fun measuring trees and tables, for climbing things inside and outside of the appropriate gyms, for balancing on wobbly lines, for the amazing questioning, personal and mathematical, for helping me appreciate doing nothing in a park as well as pushing hard in the library and for opening my mind to new things. As important as my professional goals for the future, my hope going forward is - similarly to what happened with my Italian friends - to be able to keep relationships with you as vivid as they are now, wherever I end up being.

Abstract

Weihrauch degree theory is a field of study which attempts to classify mathematical theorems based on their computational content. Brattka and Gherardi obtained a picture of the Weihrauch degrees of theorems of mathematical analysis which is stratified by the so-called choice and boundedness principles. We generalise their work to the realm of *higher descriptive set theory*, where the role of Baire space ω^ω is taken by generalized Baire space κ^κ (for uncountable cardinals κ satisfying $\kappa^{<\kappa} = \kappa$) and the role of the real line is taken by Galeotti's generalized real line $\mathbb{R}_\kappa$. To achieve this, we adapt a framework of Brattka to obtain a general theory of κ-computable metric spaces. Subsequenty, we check which of the techniques for proving non-reductions in the classical setting can be transported to the generalized setting. Lastly, we draw from the classical literature to prove several Weihrauch reducibility results in the generalized context. The result is a fairly complete picture (page 97) of the Weihrauch degrees of many of the currently known generalized analysis theorems and of the generalized choice and boundedness principles.

Contents

Introduction

Background

The study of Weihrauch degrees is an approach to the classification of mathematical theorems in terms of their computational content. The area was born with the introduction of the so-called *Type 2 Theory of Effectivity (TTE)* (see [41], [40] and [39]) which is a framework introduced by Weihrauch to transfer the notion of Type 2 computability on Baire space to sets of cardinality up to $2^{\aleph_0}$ via *representations*. These are *naming systems* associating to each element of the represented set one or more *codes* in Baire space. This allows us to study relations on represented spaces in terms of their *realizers*. Formally, if X and Y are represented spaces and $R \subseteq X \times Y$ is a relation, a partial function $f :\subseteq \omega^\omega \to \omega^\omega$ is a realizer of R if the following diagram commutes

$$\begin{tikzcd}
\omega^\omega \arrow[r, "f"] \arrow[d, "\mathrm{rep}_X"'] & \omega^\omega \arrow[d, "\mathrm{rep}_Y"] \\
X \arrow[r, "R"] & Y
\end{tikzcd}$$

in the sense that for all $x \in \mathrm{dom}(R)$ and all p coding x, $f(p)$ codes some y such that $(x, y) \in R$. Since most spaces of interest for mathematical analysis have cardinality $\leq 2^{\aleph_0}$, they can in principle be represented. Consequently, it is possible to study relations on them in terms of their realizers.

The notion of realizer then is employed to define *Weihrauch reducibility*, which intuitively captures the concept of a relation being computationally simpler than another. Formally, if $R \subseteq X \times Y$ and $S \subseteq U \times V$, we say that R *Weihrauch reduces to* S if there are computable preprocessing and postprocessing functions on Baire space such that for every $f :\subseteq \omega^\omega \to \omega^\omega$ realizing S, composing f with these pre/postprocessing functions yields a realizer of R.

$$\begin{tikzcd}
\omega^\omega \arrow[r, "\mathrm{pre}"] \arrow[dd, "\mathrm{rep}_X"'] & \omega^\omega \arrow[r, "f"] \arrow[d, "\mathrm{rep}_U"'] & \omega^\omega \arrow[r, "\mathrm{post}"] \arrow[d, "\mathrm{rep}_V"] & \omega^\omega \arrow[dd, "\mathrm{rep}_Y"] \\
 & U \arrow[r, "S"] & V & \\
X \arrow[rrr, "R"] & & & Y
\end{tikzcd}$$

Schematically, it must be the case that whenever f makes the inner diagram commute, the outer diagram commutes as well.

Now, given any theorem T of analysis which can be written as a Π_2 statement, i.e., in the form $\forall x \in X \exists y \in Y\ \varphi(x, y)$, we can consider its *associated relation* R consisting of those paris $(x, y) \in X \times Y$ such that $\varphi(x, y)$. In these terms, a realizer of the relation R is a function that sends names of elements $x \in X$ (i.e., instances of the theorem T) to names of elements $y \in Y$ which witness the truth of the theorem T. Moreover, if R is the relation associated to theorem T and S is the relation associated to theorem T', saying that R Weihrauch reduces to S means that, if we assume the existence of a black box producing names for witnesses to the truth of theorem T', we can computably obtain a similar black box for T. This is a possible formalization of the concept of a theorem being computationally stronger than another which logicians have found appealing.

Researchers then took an interest in the study of Weihrauch reducibility between various mathematical theorems. An example of this pursuit is given by the paper [7], where Brattka and Gherardi classify the *Weihrauch degrees* (i.e., the equivalence classes given by Weihrauch reducibility) of many cornerstone theorems in analysis. The following diagram, (which we will henceforth refer to as the Brattka-Gherardi diagram) is a slightly modified version of the table appearing on page 5 of [7], and it is a representation of most of the classification work contained in the above mentioned paper.[1]

$$\begin{tikzcd}
\vdots \arrow[d] & \\
C_2 \arrow[d] & \\
C_1 \equiv_{\mathrm{W}} \mathrm{B}_{\mathbb{R}} \equiv_{\mathrm{W}} \widehat{\mathrm{LPO}} \equiv_{\mathrm{W}} \mathrm{EC} \arrow[d] & \\
\mathrm{C}_{\mathrm{A}} \arrow[d] \arrow[ddr, bend left] & \\
\mathrm{C}_{\mathrm{K}} \equiv_{\mathrm{W}} \widehat{\mathrm{LLPO}} \equiv_{\mathrm{W}} \mathrm{WKL} \equiv_{\mathrm{W}} \mathrm{Sep} \arrow[d] & \\
\mathrm{C}_{\mathrm{I}} \equiv_{\mathrm{W}} \mathrm{B}_{\mathrm{I}} \equiv_{\mathrm{W}} \mathrm{IVT} \arrow[d] & \mathrm{B}_{\mathrm{I}}^{+} \arrow[l] \arrow[d] \\
\mathrm{C}_{\mathrm{I}}^{-} \equiv_{\mathrm{W}} \mathrm{B}_{\mathrm{I}}^{-} \arrow[d] & \mathrm{C}_{\omega} \equiv_{\mathrm{W}} \mathrm{B}_{\mathrm{F}} \equiv_{\mathrm{W}} \mathrm{BCT} \arrow[l] \arrow[d] \\
\mathrm{LLPO} \arrow[d] & \mathrm{LPO} \arrow[l] \\
\mathrm{id}_{\omega^{\omega}} &
\end{tikzcd}$$

In the table above, a black arrow from T to T' indicates that T' Weihrauch reduces to T, and it is complete, i.e., no arrows can be added except those that follow from transitivity of Weihrauch reducibility. We remark that the original diagram contained the degrees of many other theorems which are usually consided to be cornerstones of real analysis. The backbone of this classification is given by the so-called *choice and boundedness principles*, which are noncomputable principles concerining subsets and sequences of real numbers, and they are denoted with the letters B and C in the table.

In line with the interest in *generalized descriptive set theory* (see [24]), i.e., the study of spaces of the form κ^κ and 2^κ where κ is an uncountable cardinal satisfying $\kappa^{<\kappa} = \kappa$, the work on Weihrauch degrees has recently been expanded to higher cardinalities. In [16], Galeotti defined the real closed field $\mathbb{R}_\kappa$ for every uncountable cardinal κ satisfying $\kappa^{<\kappa} = \kappa$. These are field extensions of $\mathbb{R}$ particularly well suited to the needs of computable analysis. Also in [16], he proved that some versions of the Intermediate Value Theorem, the Extreme Value Theorem and the Baire Category Theorem hold for $\mathbb{R}_\kappa$, effectively initiating the study of *generalized analysis* (for generalised analysis see also [11], on generalizations of the Bolzano-Weierstrass Theorem to non-Archimedean fields). Moreover, in his [16] and [17], as well as with Nobrega in [19], Galeotti started the study of *generalized computable analysis*, i.e., the study of Weihrauch degrees for relations (in particular theorems) on $\mathbb{R}_\kappa$, κ^κ and related spaces.

[1]The table presented contains two principles which are not mentioned in [7], namely Sep and EC. These are included as we will use their generalized versions in our classification efforts later in the thesis. Note that the original paper uses the notation B for what we call $\mathrm{B}_{\mathbb{R}}$. All equivalences and reductions shown are either proven or referenced in [7], with the exception of those involving the newly added principles. For a proof of the equivalence EC $\equiv_{\mathrm{W}}$ $\mathrm{B}_{\mathbb{R}}$ see [39, Theorem 4] and for a proof of the equivalence Sep $\equiv_{\mathrm{W}}$ WKL see [20, Theorem 6.7].

Project overview

This thesis generalizes the choice and boundedness principles of [7][2] and it studies the Weihrauch reductions between these new principles, in the hope that they will form the backbone of the *generalized Weihrauch hierarchy*, i.e., that they will aid in classifying future generalized analysis results, similarly to how their classical counterpart capture many well-known results in analysis. The final result of the thesis is our own classification diagram containing a fairly complete picture of the Weihrauch degrees corresponding to our generalized principles and some generalized analysis results (see page 97). This project was tackled in the following way: first we investigated whether the *separation techniques* of Section 4 in [7] would carry over to the κ-context. This provided us with some tools to prove non-reducibility between principles. Subsequently, we went about the actual classification of our generalized principles. We collected all previously existing classification results from [16], [17] and [19] and we proved new classification results, sometimes drawing from the classical proofs found in the literature (see [6], [7], [39], [40], [20] and [5]). Along the way, we generalized the theory of computable metric spaces to higher versions of metric spaces, and obtained new analytical and computational results on Galeotti's generalized real line $\mathbb{R}_\kappa$.

Overview of the chapters

This thesis is organized as follows: Chapter 1 is an exposition of all the material necessary for the development of the following chapters, introducing the basic notions of generalized metric spaces, transfinite computability, (generalized) computable analysis and the surreal number fields that play a central role throughout the thesis. Chapter 2 introduces computable κ-metric spaces, which are generalizations of the classical computable metric spaces found in, e.g., [41] or [5]. One of the main results of the chapter is Theorem 2.2.2, which states that computable functions on computable κ-metric spaces are exactly those for which we can compute preimages of open sets. This result is used extensively in the rest of the thesis, and in particular it is a stepping stone for Theorem 2.3.12, which is the other most important result in the chapter. This result establishes the existence of a countable family of choice functions on generalized Baire space κ^κ which is linearly ordered with respect to Weihrauch reducibility. Moreover, it shows that each member of the family is *Weihrauch complete* for a particular set of functions. Chapter 3 contains formal definitions for the generalized versions of all principles in [7], as well as conventions on notation. The main result of this chapter is Lemma 3.1.8 which highlights a computational difference between closed sets of $\mathbb{R}$ and closed sets in $\mathbb{R}_\kappa$ based on topological differences between the two fields. Chapter 4 contains proofs for the generalized versions of the Separation Techniques of [7]. Chapter 5 contains all the Weihrauch reductions between generalized principles that we currently know of, as well as some new results on $\mathbb{R}_\kappa$ that were obtained along the way. Chapter 6 summarises our classification results and presents them in a table (page 97) analogous to the Brattka-Gherardi diagram. In it, we compare the two diagrams and explicitly state the questions which remain open.

Background knowledge and notation

Throughout the thesis, we assume familiarity with the basic notions of set theory. We refer the reader to [23] for general set theory and to [15] for more about Gödel's constructible universe L. Moreover, we assume familiarity with classical computability theory, and we refer the reader to [31] for more on the topic. All the material we will need about transfinite computability is explicitly introduced, and it is taken from [10]. We introduce computable analysis in the generalized setting directly, for more on classical computable analysis the standard reference is [41]. For operations on Weihrauch degrees,

[2] We give reference to the definitions of the generalised principles corresponding to those in the Brattka-Gherardi diagram: the analogues of the functions C_n at the top of the diagram are defined on page 55, the generalized versions of the choice principles, indicated by the letter C, the boundedness principles, indicated by the letter B and the omniscience principles LPO and LLPO are defined in Section 3.2. Section 3.2 also contains definitions of the generalized counterparts to WKL and Sep, as well as the generalized analysis results analogous to those indicated by IVT and BCT and other auxiliary principles which are necessary for the classification in the generalized context.

we refer to [6].

We use *greek letters* $\alpha, \beta, \gamma \dots$ to denote ordinals, whereas we reserve the letter κ for cardinals. We sometimes use the notation $|X|$ to denote the *cardinality* of the set X (also if X is an ordinal). Given a set X and a subset $A \subseteq X$, we use the notation A^{c} to denote the complement of A in X, i.e., $X \setminus A$. We will only use this notations in settings where the ambient set X is clear from the context. For any well-ordered set $(X, <)$, we use the notation $\operatorname{ot} X$ to refer to the unique ordinal isomorphic to it. If X, Y denote sets, we write X^Y to refer to the *set of functions* from Y to X. We use the same exponential notation also for *ordinal exponentiation* and *cardinal exponentiation*: this will not be an issue as the intended meaning of our symbols is always disambiguated either by the context or by explicit mention. For any set X and any ordinal β, we write $X^{<\beta}$ as a shorthand for the set $\bigcup_{\alpha\in\beta} X^\alpha$. Accordingly, $X^{\leq\beta}$ denotes the set $X^{<\beta+1}$. If α and β are ordinals, we sometimes call an element s of α^β a *word*. We call the domain of a word s its *length* and we denote it as $l(s)$ or $|s|$. We denote the *concatenation* of two words s and w, defined in the obvious way, as $s^\frown w$ or sw. Given a sequence of words $(w_\alpha)_{\alpha\in\gamma}$, we denote their concatenation as $[\![w_\alpha]\!]_{\alpha\in\gamma}$. Given a subset A of a set X, we denote by χ_A its *characteristic function*, i.e., the function $\chi_A \colon X \to 2$ such that $\chi_A(x) = 1$ if and only if $x \in A$.

Given a totally ordered set $(I, <)$, we say that $A \subseteq I$ is an *interval* if it is convex, i.e., for every $a, b \in A$ if $a < c < b$, then $c \in I$. Moreover, if $a < b$ we denote as (a, b) the *open* interval determined by endpoints a and b, i.e., $(a, b) = \{c \in I \mid a < c < b\}$. Accordingly we denote the *closed* interval as $[a, b] = (a, b) \cup \{a, b\}$, and the *half-closed* intervals as $[a, b) = (a, b) \cup \{a\}$ and $(a, b] = (a, b) \cup \{b\}$. Lastly, if I does not have a maximum, we denote as $(a, +\infty)$ the interval given by $\{c \in I \mid a < c\}$. The symbols $[a, +\infty)$, $(-\infty, a)$ and $(-\infty, a]$ are defined analogously. If A and B are subsets of the totally ordered set $(I, <)$, we write $A < B$ as a shorthand for $\forall a \in A \forall b \in B\ (a < b)$. If $A = \{x\}$ is a singleton, we write $x < B$ instead of $\{x\} < B$.

Chapter 1

Preliminaries

1.1 Topology

Topology is a fundamental tool in the development of computable analysis. We give an account of the notions which will be used in the thesis. We assume familiarity with basic topological notions which can be found, for example, in [42] or [29].

1.1.1 λ-metric spaces

For any uncountable regular cardinal λ, it is possible to define the notion of λ-metric spaces, which are sets equipped with a distance function which has values in an ordered group admitting strictly decreasing sequences of positive elements of length λ. These spaces were first introduced by Sikorski (see for example [37]) and their theory was recently expanded by Galeotti in [16] and [17], Motto Ros, Agostini and Schlicht in [1] and Schlicht and Coskey in [14], in the contexts of generalized descriptive set theory and generalized analysis.

Definition 1.1.1 (Base number)**.** *Given a* totally ordered abelian group $(G, +, 0 \leq)$*, let* $G^+ = \{x \in G \mid x > 0\}$ *be the* positive part *of* G*. We will call* $\mathrm{coi}(G^+)$ *the* base number *of* G*, i.e.,the least possible cardinality of a set* Y *which is coinitial in* G^+*. We denote the base number of* G *as* $\mathrm{bn}(G)$*.*

Lemma 1.1.2. *Let* $(A, \leq)$ *be a linearly ordered set with no least element, and let* λ *be its coinitiality, then* λ *is regular.*

Proof. Follows from a proof analogous to [23, Lemma 3.8]. $\square$

Definition 1.1.3 (G-metric)**.** *Let* $(G, +, 0, \leq)$ *be a totally ordered abelian group, and let* X *be a set. A function* $d\colon X \times X \to G$ *is called a* G-metric *if*

- *for every* $x, y \in X$*,* $d(x, y) \geq 0$ *and* $d(x, y) = 0 \iff x = y$*,*
- *for every* $x, y \in X$*,* $d(x, y) = d(y, x)$*,*
- *for every* $x, y, z \in X$*,* $d(x, y) + d(y, z) \geq d(x, z)$*.*

Given any totally ordered abelian group G, we refer to a pair (X, d), with X a set and $d\colon X \times X \to G$ a G-metric as a *G-metric space.*

Definition 1.1.4 (Open and closed balls)**.** *Let* G *be a totally ordered group and let* $\mathfrak{X} = (X, d)$ *be a* G*-metric space. We define the* open balls *of* d *as the sets of the form*

$$B_{\mathfrak{X}}(x, g) = \{x' \in X \mid d(x, x') < g\}$$

and the closed balls *of* d *as the sets of the form*

$$\overline{B}_{\mathfrak{X}}(x, g) = \{x' \in X \mid d(x, x') \leq g\}.$$

Given any totally ordered abelian group G and any set X, a G-metric $d\colon X \times X \to G$ induces a topology τ_d on X consisting of the smallest collection containing the base $\mathcal{B}_d = \{B_{\mathfrak{X}}(x,g) \mid x \in X, g \in G\}$ which is closed under arbitrary unions. We call τ_d the *topology induced* by d.

A related concept which features prominently in generalized analysis is that of a λ-*topology*: given a cardinal λ and a set X, a λ-topology on X is a subset of $\mathcal{P}(X)$ which contain the empty set, the whole space X and it is closed under finite intersections and unions indexed by ordinals below λ. Similarly to the situation for ordinary topologies, given a set $B \subseteq \mathcal{P}(X)$ closed under finite intersections and such that $\bigcup B = X$,[1] we define the λ-topology generated by B as the smallest subset of $\mathcal{P}(X)$ which contains the empty set and the whole space X, includes B and is closed under unions indexed by ordinals below λ. Given a cardinal λ, a totally ordered abelian group G and a G-metric space (X,d), it is clear from the definitions that, if we denote with τ_d the topology induced by d and with τ_d^λ the λ-topology induced by d, then we have $\tau_d^\lambda \subseteq \tau_d$.

Definition 1.1.5 (λ-metrizability)**.** *A topological space (X,τ) is called* λ-metrizable *if there exists a totally ordered abelian group G with* $\mathrm{bn}(G) = \lambda$ *and a G-metric d on X such that τ is the topology induced by the metric d.*

Similarly, given a pair (X,d) where d is a G-metric for some group G with $\mathrm{bn}(G) = \lambda$, call (X,d) a λ-*metric space*. In the rest of this section, whenever unspecified, assume that the symbols G and H stand for totally ordered additive abelian groups.

In light of Lemma 1.1.2, whenever we speak of λ-metrizable (λ-metric) spaces in the rest of this thesis, we will assume that λ is an uncountable regular cardinal.

Definition 1.1.6 (λ-additivity)**.** *Given an uncountable cardinal λ, a topological space (X,τ) is called* λ-additive *if, for all $\delta < \lambda$ and all δ-sequences of open sets $(A_\alpha)_{\alpha\in\delta}$, we have that $\bigcap_{\alpha\in\delta} A_\alpha$ is open.*

It is known that for any regular cardinal λ, the notion of λ-metrizability can be characterized by the topological property of λ-additivity.

Proposition 1.1.7 ([36], Theorem viii)**.** *Let λ be a regular uncountable cardinal, let (X,τ) be a topological space which admits a base of cardinality $\leq\lambda$ and let G be a totally ordered abelian group with* $\mathrm{bn}(G) = \lambda$. *The space (X,τ) is λ-additive if and only if it is G-metrizable.*

In particular, this means that if (X,τ) is a topological space with a base of size λ and G and G' are two totally ordered abelian groups with base number λ, then (X,τ) is G-metrizable if and only if it is G'-metrizable. This equivalence allows us to prove results on general λ-metrizable spaces while assuming that we are working with specific any specific group G.

We mention here that this is in stark contrast with the situation for $\aleph_0$-metric spaces. The latter category obviously includes ordinary metric spaces, but a result analogous to Proposition 1.1.7 does not hold. In fact, it is easy to see that no connected metric space with more than two points is $\mathbb{Q}$ metrizable.[2]

It is immediate to see that for a function $f\colon (X,d_X) \to (Y,d_Y)$ where d_X is a G-metric and d_Y is an H-metric, topological continuity (the requirement that preimages of open sets in (Y,d_Y) via f are open in (X,d_X)) coincides with the usual ϵ-δ definition of continuity in metric spaces, i.e., f is continuous if and only if

$$\forall x \in X \forall \epsilon \in G^+ \exists \delta \in H^+ \forall y \in X \left(d_X(x,y) \leq \delta \implies d_Y(f(x),f(y)) \leq \epsilon\right).$$

Similarly we can modify the definition of uniform continuity for functions between metric spaces:

[1] In line with standard terminology, we refer to sets satisfying these two requirements as topological bases on X.

[2] We give a sketch of how one could prove this: let (X,d) be a connected metric space and let x and y be distinct points in X. We show that $\mathrm{ran}(d) \not\subseteq \mathbb{Q}$ by contradiction. So, assume $\mathrm{ran}(d) \subseteq \mathbb{Q}$ and let $r > 0$ be an irrational number such that $r < d(x,y)$. Consider the closed ball $B = \overline{B}_X(x,r)$: this is a closed set as all closed balls in metric spaces are closed. On the other hand, the assumption $\mathrm{ran}(d) \subseteq \mathbb{Q}$ implies that $B = \bigcup_{q\in\mathbb{Q},q<r} B(x,q)$. So, B is also open, and in particular it is a nonempty clopen set different from X. This contradicts the connectedness of X. This immediately implies that d is not equivalent to any $\mathbb{Q}$-metric and proves the claim.

Definition 1.1.8 (Uniform continuity)**.** *Let (X, τ_X) and (Y, τ_Y) be two topological spaces and let $d_X : G \to X$ and $d_Y : H \to Y$ be metrics which induce the respective topologies. A function $f : X \to Y$ is called* uniformly continuous *if*

$$\forall \epsilon \in G^+ \exists \delta \in H^+ \forall x, y \in X \left(d_X(x, y) \leq \delta \implies d_Y(f(x), f(y)) \leq \epsilon \right).$$

We contrast these notions with an alternative concept of continuity, which is more relevant for generalized analysis (see, e.g., [16] and [2]).

Definition 1.1.9. *Let λ be a cardinal, let X and Y be sets, let τ and τ' be λ-topologies on X and Y, respectively. Let $f : X \to Y$ be a function. We say that f is λ-continuous if for every $U \in \tau'$, $f^{-1}[U] \in \tau$.*

We mention that, having fixed topological bases B_X and B_Y, λ-continuity is a stronger property than continuity (see, e.g., [16, Lemma 3.2.4] for a proof).

Lemma 1.1.10. *Let X and Y be sets, B_X and B_Y be topological bases on X and Y respectively. Let τ_X be the topology on X induced by B_X and let τ_Y be the topology on Y induced by B_Y. Similarly, call τ_X^λ and τ_Y^λ the λ-topologies induced by B_X and B_Y. Let $f : X \to Y$ be λ-continuous with respect to τ_X^λ and τ_Y^λ, then f is continuous with respect to τ_X and τ_Y.*

We fix some more terminology which will be useful in the rest of the thesis.

Definition 1.1.11 (Sequences)**.** *Let α be an ordinal and let X be any set, we call a function $s : \alpha \to X$ an* α-sequence *(or just a* sequence *if its domain is clear from the context). For any $\beta \in \alpha$, we often write s_β instead of $s(\beta)$. Accordingly, we write $(s_\beta)_{\beta\in\alpha} \subseteq X$ instead of $s : \alpha \to X$.*

Definition 1.1.12 (Convergent and Cauchy sequences)**.** *Let (X, d) be a G-metric space where G is a totally ordered abelian group with* $\mathrm{bn}(G) = \lambda$ *and let $s : \lambda \to X$ be a sequence. We say that s is* Cauchy *if*

$$\forall \epsilon \in G^+ \exists \alpha \in \lambda \forall \beta, \gamma > \alpha \left(d(s_\beta, s_\gamma) < \epsilon \right).$$

We say that s converges to $x \in X$ *if*

$$\forall \epsilon \in G^+ \exists \alpha \in \lambda \forall \beta > \alpha \left(d(s_\beta, x) < \epsilon \right).$$

We denote this as $\lim_{\alpha\to\lambda} s_\alpha = x$.

As in the classical context, every convergent sequence is easily seen to be Cauchy (for example see the proof sketch under [42, Definition 24.1]), but the converse does not hold.

Definition 1.1.13 (Cauchy completeness)**.** *Let (X, d) be a G-metric space where G is a totally ordered abelian group with* $\mathrm{bn}(G) = \lambda$*, we say that X is* Cauchy complete *(or just* complete*) if every Cauchy sequence in X converges to some element of X.*

Definition 1.1.14 (λ-separability)**.** *Given an uncountable cardinal λ, a topological space (X, τ) is called* λ-separable *if it has a dense subset of cardinality λ.*

Next we want to settle some terminology and state some facts about the interplay between sequences, topological spaces and continuous functions.

Definition 1.1.15 (Sequential continuity)**.** *Let X and Y be topological spaces, a function $f : X \to Y$ is* λ-sequentially continuous *if for every λ-sequence $(x_\alpha)_{\alpha\in\lambda}$ of elements of X which converges, we have* $\lim_{\alpha\to\lambda}(f(x_\alpha)) = f(\lim_{\alpha\to\lambda} x_\alpha)$.

Definition 1.1.16 (λ-compactness)**.** *A topological space (X, τ) is called λ-compact if every open cover $\mathcal{U} = \{U_i \mid i \in I\}$ admits a refinement given by $<\lambda$ sets, i.e.,some set $J \subseteq I$ such that $|J| < \lambda$ and $\{U_j \mid j \in J\}$ is a cover.*

The following result is analogous to the corresponding result for the classical context, and it is proved in the same way.

Proposition 1.1.17. *Let (X, τ) be a λ-compact topological space and let $C \subseteq X$ be a closed set. Then C is λ-compact. Conversely, if (Y, ξ) is any Hausdorff and λ-additive topological space, and $D \subseteq Y$ is any λ-compact set, then D is closed.*

Proof. See Theorem 26.2 and Theorem 26.3 in [29] for proofs of the corresponding statements for ordinary compact spaces. The proof of Theorem 26.2 can be carried over to our context with no modifications, whereas repeating the proof of Theorem 26.3 in our context requires the extra assumption of λ-additivity. Note that any λ-metric space is both Hausdorff and λ-additive. □

Definition 1.1.18 (λ-sequential spaces and λ-sequentially compact spaces)**.** *A topological space (X, τ) is called* λ-sequential *if for every $A \subseteq X$, it holds that*

$$\mathrm{cl}(A) = \{y \in X \mid \exists (a_\alpha)_{\alpha \in \lambda} \subseteq A(\lim_{\alpha \to \lambda} a_\alpha = y)\},$$

and it is called λ-sequentially compact *if and only if every λ-sequence in X has a convergent subsequence.*

The following results correspond to well known theorem for metric spaces, we give references to the proofs of the classical theorems, noting that they all transfer without modifications.

Theorem 1.1.19. *Let λ be any uncountable regular cardinal and let (X, d) be a λ-metric space with d a G-metric. Then (X, d) is λ-sequential. Moreover, if X is λ-compact, then X is λ-sequentially compact.*

Proof. See [42, Theorem 10.4] and [29, Theorem 28.2]. □

Theorem 1.1.20. *Let (X, τ) and (Y, ξ) be topological spaces with X a λ-sequential space and let $f\colon X \to Y$ be a function. Then f is continuous if and only if f is λ-sequentially continuous.*

Proof. See [42, Corollary 10.5]. □

This immediately yields a λ-version of Heine's theorem.

Theorem 1.1.21 (λ-Heine's Theorem)**.** *Let (X, d_X) and (Y, d_Y) be λ-metric spaces with d_X a G-metric and d_Y an H-metric, let X be λ-sequentially compact and let $f\colon X \to Y$ be a continuous function. Then f is uniformly continuous.*

Proof. See [35, Theorem 4.19]. □

We mention that a version of the Baire Category Theorem holds for generalized metric spaces:

Definition 1.1.22 (λ-spherical completeness)**.** *A topological space (X, τ) is called* λ-spherically complete[3] *if for every ordinal $\beta < \lambda$, every sequence of open balls $(A_\alpha)_{\alpha \in \beta}$ such that, for all $\delta < \gamma$, $A_\gamma \subseteq A_\delta$, has nonempty intersection.*

Theorem 1.1.23 (λ-Baire Category Theorem)**.** *Let (X, d) be a λ-metric, λ-spherically complete and Cauchy complete space and let $(A_\alpha)_{\alpha \in \lambda}$ be a sequence of closed sets with empty interior, then $\bigcup_{\alpha \in \lambda} A_\alpha$ has empty interior.*

Proof. See [16], Theorem 3.5.24. □

Given a cardinal λ we introduce λ-Borel sets (this definition can be found in [16]):

Definition 1.1.24 (λ-Borel sets)**.** *Let $\mathfrak{X} = (X, d)$ be a λ-metric space. Define $\mathcal{B}(\mathfrak{X})$, the λ-Borel subsets of X, as the smallest subset of $P(X)$ which contains the open sets and is closed under unions of size λ and complementation.*

As in the classical case, the λ-Borel sets are stratified in a hierarchy given by:

[3]Note that in the literature the notion of λ-spherical completeness is also sometimes called $<\lambda$-spherical completeness.

- $\mathbf{\Sigma}^0_1(\mathfrak{X})$ = open sets,
- $\mathbf{\Pi}^0_1(\mathfrak{X})$ = closed sets,
- for every $\alpha > 1$, $\mathbf{\Sigma}^0_\alpha(\mathfrak{X}) = \{\bigcup_{\beta<\lambda} A_\beta \mid \forall\beta\,\exists\gamma < \alpha\,(A_\beta \in \mathbf{\Pi}^0_\gamma(\mathfrak{X}))\}$,
- for every $\alpha > 1$, $\mathbf{\Pi}^0_\alpha(\mathfrak{X}) = \{X \setminus A \mid A \in \mathbf{\Sigma}^0_\alpha(\mathfrak{X})\}$,
- for every $\alpha \geq 1$, $\mathbf{\Delta}^0_\alpha(\mathfrak{X}) = \mathbf{\Sigma}^0_\alpha(\mathfrak{X}) \cap \mathbf{\Pi}^0_\alpha(\mathfrak{X})$.

It can be shown that this is a proper hierarchy of length λ^+ when $\lambda^{<\lambda} = \lambda$ and $X = \lambda^\lambda$ or $X = 2^\lambda$ ([16, Theorem 3.5.21]).

We prove a couple of results related to Borel sets on λ-metric spaces which will be useful for our treatment of computable κ-metric spaces.

Proposition 1.1.25. *Let $\mathfrak{X} = (X, d)$ be a λ-metric space for some cardinal λ satisfying $\lambda^{<\lambda} = \lambda$. For all $n \in \omega$, $\mathbf{\Sigma}^0_n(\mathfrak{X})$ is closed under intersections of size $<\lambda$. Similarly, $\mathbf{\Pi}^0_n(\mathfrak{X})$ is closed under unions of size $<\lambda$.*

Proof. We prove the claim by induction on n: by Proposition 1.1.7, the claim holds for $\mathbf{\Sigma}^0_1$. Note that by a simple application of the De Morgan Laws, the claim for $\mathbf{\Pi}^0_n$ follows from the one for $\mathbf{\Sigma}^0_n$. Therefore we only need to prove that, assuming closure of the $\mathbf{\Sigma}^0_n$ sets under small intersections (and therefore closure of the $\mathbf{\Pi}^0_n$ sets under small unions) we obtain that $\mathbf{\Sigma}^0_{n+1}$ sets are closed under small intersections. To this end, let $(A_\alpha)_{\alpha\in\delta}$ be a sequence of $\mathbf{\Sigma}^0_{n+1}$ sets, then for every α there are sets $(C^\alpha_\beta)_{\beta\in\lambda}$ in $\mathbf{\Pi}^0_n$ such that $\bigcup_{\beta\in\lambda} C^\alpha_\beta = A_\alpha$. Now, for every $f \in \lambda^\delta$ and every $\alpha \in \delta$, let $C^\alpha_f = C^\alpha_{f(\alpha)}$, then

$$\bigcap_{\alpha\in\delta} A_\alpha = \bigcap_{\alpha\in\delta} \bigcup_{\beta\in\lambda} C^\alpha_\beta = \bigcup_{f\in\lambda^\delta} \bigcap_{\alpha\in\delta} C^\alpha_f,$$

Since $\mathbf{\Pi}^0_n$ is closed intersections of size λ, each $\bigcap_{\alpha\in\delta} C^\alpha_f$ is a $\mathbf{\Pi}^0_n$-set, and since $|\lambda^\delta| = \lambda$, this shows that $\bigcap_{\alpha\in\delta} A_\alpha$ can be written as a union of λ-many $\mathbf{\Pi}^0_n$ sets, i.e.,it belongs to $\mathbf{\Sigma}^0_{n+1}$. This concludes the induction and thus proves the claim. $\square$

Definition 1.1.26 ($\mathbf{\Sigma}^0_\alpha$-measurability)**.** *Let $\mathfrak{X} = (X, d)$ be a λ-metric space and let (Y, ξ) be any topological space. Let $f :\subseteq X \to Y$ be a partial function and let α be an ordinal, then f is called $\mathbf{\Sigma}^0_\alpha$-*measurable *if for all open sets $O \in \xi$, there exists some $V_O \in \mathbf{\Sigma}^0_\alpha(\mathfrak{X})$ such that $f^{-1}[O] = V_O \cap \mathrm{dom}(f)$.*[4] *We denote the set of $\mathbf{\Sigma}^0_\alpha$-measurable functions from X to Y as $\mathrm{Meas}_\alpha(\mathfrak{X}, Y)$. If $(Y, \xi) = (X, \mathbf{\Sigma}^0_1(\mathfrak{X}))$, we just write $\mathrm{Meas}_\alpha(\mathfrak{X})$.*

Proposition 1.1.27. *Let $\mathfrak{X} = (X, d)$ be a λ-metric space and let α be an ordinal such that $\mathbf{\Sigma}^0_\alpha(\mathfrak{X}) \neq \mathbf{\Sigma}^0_{\alpha+1}(\mathfrak{X})$ and let (Y, ξ) be any T_0 topological space with at least two points, then $\mathrm{Meas}_{\alpha+1}(\mathfrak{X}, Y) \neq \mathrm{Meas}_\alpha(\mathfrak{X}, Y)$.*

Proof. Let $A \in \mathbf{\Sigma}^0_{\alpha+1}(\mathfrak{X}) \setminus \mathbf{\Sigma}^0_\alpha(\mathfrak{X})$ and let y_1 and y_2 be distinct points in Y and define the function $f\colon X \to Y$ as

$$f(x) = \begin{cases} y_1 & \text{if } x \in A \\ y_2 & \text{if } x \notin A \end{cases}$$

Now if $O \in \xi$ is any open set, $f^{-1}[O]$ can either be X, $\emptyset$ or A, and all three options are $\mathbf{\Sigma}^0_{\alpha+1}(\mathfrak{X})$, hence f is $\mathbf{\Sigma}^0_{\alpha+1}$-measurable. On the other hand, since Y it T_0 and $y_1 \neq y_2$, there exists an open set O' with $y_1 \in O'$ and $y_2 \notin O'$, hence $f^{-1}[O'] = A$. Since $A \notin \mathbf{\Sigma}^0_\alpha(\mathfrak{X})$, it follows that f is not $\mathbf{\Sigma}^0_\alpha$-measurable, proving the claim. $\square$

Corollary 1.1.28. *Let $\lambda = \lambda^{<\lambda}$ and let $X \in \{2^\lambda, \lambda^\lambda\}$, then for every $\alpha < \lambda^+$, $\mathrm{Meas}_\alpha(X) \neq \mathrm{Meas}_{\alpha+1}(X)$.*

[4] Clearly if f is total, then $V_O = f^{-1}[O]$.

1.1.2 Generalized Baire and Cantor spaces and trees

We introduce generalizations of Baire space and Cantor space to uncountable cardinals. These will play the roles of Baire and Cantor space in our generalized computability framework, and they are important in generalized computable analysis. We mention that these spaces are of great interest in the area of generalized descriptive set theory. Definitions and results in this section are either folklore, from [17] or from [1]. We refer the interested reader to the mentioned papers for more details. Throughout this subsection, we work with the standard assumption that κ is an uncountable cardinal satisfying $\kappa^{<\kappa} = \kappa$.

Definition 1.1.29 (Initial segments and basic opens)**.** *Let $\theta \in \{2, \kappa\}$ and let $s \in \theta^{<\kappa}$ and $t \in \theta^{<\kappa} \cup \theta^{\kappa}$ be two sequences:* s is an initial segment of t *if* $\mathrm{dom}(s) \subseteq \mathrm{dom}(t)$ *and for all* $\alpha \in \mathrm{dom}(s)$*,* $s(\alpha) = t(\alpha)$*. We will denote this as $s \subseteq t$. For every $s \in \theta^{<\kappa}$, define:*

$$[s] = \{x \in \theta^{\kappa} \mid s \subseteq x\}.$$

Definition 1.1.30 (Generalized Baire and Cantor spaces.)**.** *Let κ be an uncountable cardinal such that $\kappa^{<\kappa} = \kappa$. Define generalized Baire space as κ^{κ} equipped with the topology generated by the basis*

$$\mathcal{B} = \{[s] \colon s \in \kappa^{<\kappa}\}.$$

Similarly define generalized Cantor space as 2^{κ} equipped with the topology generated by the basis

$$\mathcal{C} = \{[s] \colon s \in 2^{<\kappa}\}.$$

We list some basic results on generalized Baire and Cantor space from the literature which we will need in the rest of the thesis.

Theorem 1.1.31. *The following holds:*

(a) the topologies on generalized Baire space κ^{κ} and generalized Cantor space 2^{κ} are κ-metrizable, therefore these spaces are κ-additive,

(b) generalized Baire space κ^{κ} and generalized Cantor space 2^{κ} are κ-spherically complete,

(c) the κ-Baire Category Theorem holds for κ^{κ} and 2^{κ}.

Proof. For the κ-metrizability of the topologies on generalized Baire and Cantor space, see page 29 of [17]. Their κ-additivity then follows from Proposition 1.1.7. For item (b) see, e.g., [1, Theorem 2.3.1]. Item (c) follows from item (b) and Theorem 1.1.23. □

We introduce the notion of trees, these objects are deeply tied to (generalized) Baire and Cantor spaces and we will use their properties in many inductive proofs throughout the thesis.

Definition 1.1.32 (Trees)**.** *Let $\theta \in \{2, \kappa\}$, a* tree over θ *is a set $T \subseteq \theta^{<\kappa}$ which is closed under initial segments, i.e.,if $s \subseteq t$ and $t \in T$, then $s \in T$.*

Note that trees are partial orders, where the relation is $\subseteq$. Given a tree T and an ordinal $\alpha \in \kappa$, we denote by T_{α} the *α-th level* of T, i.e.

$$T_{\alpha} = \{t \in T \mid |t| = \alpha\}.$$

Given a tree T, we say that T has *height* γ, we will denote this as $h(T) = \gamma$, if $\gamma = \sup\{\alpha \in \kappa \mid T_{\alpha} \neq \emptyset\}$. Notice that generalized Cantor spaces can be seen as a subspaces of generalized Baire spaces. Moreover, the classical characterization of the topology applies (see [1], pg 9):

Proposition 1.1.33. *Let $\theta \in \{2, \kappa\}$, then a set $C \subseteq \theta^{\kappa}$ is closed if and only if there exists a tree $T \subseteq \theta^{<\kappa}$ such that $C = [T]$ where*

$$[T] = \{x \in \theta^{\kappa} \mid \forall \alpha \in \kappa \ \ x{\restriction}\alpha \in T\}.$$

$[T]$ is called the set of branches of T.

Following [1], we say that a tree $T \subseteq \theta^\kappa$ is *$<\kappa$-closed* if for all limit ordinals $\gamma \in \kappa$ and all $x \in \theta^\kappa$, if $x \restriction \beta \in T$ for all $\beta \in \gamma$, then $x \in T$.

We define the notion of the extendible part of a tree:

Definition 1.1.34 (Extendible part). *Let $T \subseteq \theta^{<\kappa}$ be a tree, define the* extendible part *of T as*

$$\mathrm{ext}(T) = \{t \in T \mid \exists x \in [T]\ (t \subseteq x)\}.$$

Intuitively, $\mathrm{ext}(T)$ is the set of those nodes that lie on branches of T. We will call elements of $\mathrm{ext}(T)$ the *extendible nodes* of T.

Lemma 1.1.35. *Let T be a tree such that $[T] \neq \emptyset$, then $\mathrm{ext}(T)$ is closed under initial segments, hence it is itself a tree.*

Proof. Follows from the definition of $\mathrm{ext}(T)$. $\square$

Note that for uncountable cardinals κ, finding a branch in a tree of height κ is considerably harder than finding a branch in a tree of height ω. We illustrate this with some examples: first we show that a simple recursive construction is enough to build branches on trees T of height ω.

Example 1.1.36. *Let $T \subseteq \omega^{<\omega}$ be a tree such that $[T] \neq \emptyset$, equivalently $\mathrm{ext}(T) \neq \emptyset$. Clearly the root of T, namely the empty sequence, is an element of $\mathrm{ext}(T)$. We can then build an element of $[T]$ recursively as follows: fix any choice function f on $\mathcal{P}(T) \setminus \emptyset$. Let $r_0 = \emptyset$ and for every n, let*

$$r_{n+1} = f(A_n) \ \textit{where}\ A_n = \mathrm{ext}(T) \cap \{t \in T_{n+1} \mid r_n \subseteq t\}$$

Note that by a straightforward induction we can prove that for every $n \in \omega$, $r_n \in \mathrm{ext}(T)$ and consequently that some of its immediate successors must be in $\mathrm{ext}(T)$ as well, therefore the sets A_n are always nonempty. The idea of this recursive definition is that we build a sequence of nodes with strictly increasing length $(r_n)_{n\in\omega}$ such that $r_n \subseteq r_{n+1}$ for all n, therefore $\bigcup_{n\in\omega} r_n \in [T]$.

Now we show that the technique above fails for trees of uncountable height:

Example 1.1.37. *Let $T = \{t \in 2^{<\kappa} \mid 0^\omega \not\subseteq t\}$. Clearly $[T] \neq \emptyset$. Now notice that if $r \in \mathrm{ext}(T)$, the set $A_r = \mathrm{ext}(T) \cap \{t \in T_{|r|+1} \mid r \subseteq t\}$ has at most two elements, more precisely $A_r \subseteq \{r^\frown 0, r^\frown 1\}$. Fix a choice function f on the set $\{A_r \mid r \in \mathrm{ext}(T)\}$ which behaves as follows: if $r^\frown 0 \in A_r$, $f(A_r) = r^\frown 0$, otherwise $f(A_r) = r^\frown 1$. Define a sequence r_α recursively as in Example 1.1.36: $r_0 = \emptyset$ and for every α, $r_{\alpha+1} = f(A_{r_\alpha})$. If γ is a limit ordinal $r_\gamma = \bigcup_{\alpha\in\gamma} r_\alpha$. It is immediate to see that the sequence can be defined up to level ω, in particular for all $\alpha \leq \omega$, $r_\alpha = 0^\alpha$, but $r_{\omega+1}$ cannot be defined as $0^\omega \notin \mathrm{ext}(T)$. This shows that the recursive procedure described above is not enough to build branches on trees of uncountable height.*

The missing ingredient in the construction of Example 1.1.37 is that, unlike in the case of trees of height ω, we cannot make sure that the nodes on our recursively built sequence are always extendible, in particular this requirement does not hold in general for limit ordinal heights. This suggests the following:

Lemma 1.1.38. *Let $T \subseteq \theta^{<\kappa}$ be a tree with $[T] \neq \emptyset$ such that $\mathrm{ext}(T)$ is $<\kappa$-closed. Let $f\colon \mathcal{P}(T)\setminus\emptyset \to T$ be any choice function and let $(r_\alpha)_{\alpha\in\kappa}$ be the sequence built by recursion as*

$$r_0 = \emptyset,\ r_{\alpha+1} = f(A_{r_\alpha}),\ r_\gamma = \bigcup_{\alpha\in\gamma} r_\alpha \ \textit{for all limit ordinals}\ \gamma$$

where $A_{r_\alpha} = \mathrm{ext}(T) \cap \{t \in T_{\alpha+1} \mid r_\alpha \subseteq t\}$.
Then the sequence $(r_\alpha)_{\alpha\in\kappa}$ is well-defined and such that $\bigcup_{\alpha\in\kappa} r_\alpha \in [T]$, in particular, $(r_\alpha)_{\alpha\in\kappa} \subseteq \mathrm{ext}(T)$.

Proof. First, it is clear by construction that for all ordinals $\beta \in \kappa$, if r_β can be defined, then $|r_\beta| = \beta$, moreover if $\alpha < \beta$, then $r_\alpha \subseteq r_\beta$. This implies that in order to prove the claim we only need to show that for every $\beta < \kappa$, r_β can actually be defined by the recursion above, i.e.,that all sets A_{r_α} are nonempty. It is immediate to see that $A_{r_\alpha} \neq \emptyset$ if and only if $r_\alpha \in \operatorname{ext}(T)$. So, our claim reduces to $r_\alpha \in \operatorname{ext}(T)$ for all $\alpha \in \kappa$. We show this by induction on α: first note that $\operatorname{ext}(T) \neq \emptyset$, hence $r_0 = \emptyset \in \operatorname{ext}(T)$. If α is a successor ordinal, then $\alpha = \beta + 1$ for some ordinal β and by inductive assumption $r_\beta \in \operatorname{ext}(T)$. So, r_α can be defined by the recursion given and by definition $r_\alpha \in A_{r_\beta} \subseteq \operatorname{ext}(T)$. Lastly if α is a limit ordinal and by inductive assumption we have defined the sequence $(r_\beta)_{\beta\in\alpha}$ with $r_\beta \in \operatorname{ext}(T)$ for all $\beta \in \alpha$, then since $r_\alpha = \bigcup_{\beta\in\alpha} r_\beta$, it holds that for all initial segments t of r_α, $t \in \operatorname{ext}(T)$. Since $\operatorname{ext}(T)$ is $<\kappa$-closed, it follows that $r_\alpha \in \operatorname{ext}(T)$. This concludes the induction and proves the claim. □

We will use this lemma as follows: whenever we have a tree T such that $[T] \neq \emptyset$ and $\operatorname{ext}(T)$ is $<\kappa$-closed, and a function $f :\subseteq T \to T$ such that $\operatorname{ran}(f) \subseteq \operatorname{dom}(f)$, $\emptyset \in \operatorname{dom}(f)$ and for all $t \in T$, $t \subseteq f(t)$ and $|f(t)| = |t|+1$, then as soon as we can prove that for all $t \in \operatorname{dom}(f)$, $t \in \operatorname{ext}(T) \implies f(t) \in \operatorname{ext}(T)$, we obtain that f determines a branch $(t_\alpha)_{\alpha\in\kappa}$ given by

$$t_0 = \emptyset,\ t_{\alpha+1} = f(t_\alpha),\ t_\gamma = \bigcup_{\alpha\in\gamma} t_\alpha \text{ for all limit ordinals } \gamma.$$

In Section 4.3 we will prove some results which rely on the property of κ-compactness of generalized Cantor space 2^κ. We state here a well known equivalent condition for the κ-compactness of 2^κ, namely the *weak compactness* of the cardinal κ. Recall that a cardinal κ has the *tree property* if for every tree T of height κ, cardinality κ and such that for all $\alpha < \kappa$, $|T_\alpha| < \kappa$, it is the case that $[T] \neq \emptyset$, and κ is *strongly inaccessible* if $2^\lambda < \kappa$ for every cardinal $\lambda < \kappa$. A cardinal κ is weakly compact if it is strongly inaccessible and it has the tree property.

Theorem 1.1.39. *Generalized Cantor space 2^κ is κ-compact if and only if κ is weakly compact.*

Proof. See, e.g., [28], Theorem 5.6. □

Corollary 1.1.40. *Let κ be a weakly compact cardinal, let $C \subseteq 2^\kappa$ a closed set and let $f : C \to (Y, d_Y)$ be a continuous function. Then f is uniformly continuous.*

Proof. Follows from Theorem 1.1.39, Proposition 1.1.17 and Theorem 1.1.21. □

1.2 Transfinite computability

The notion of computability is central in this thesis. Since our objective is to generalize some results from the theory of computable analysis to uncountable cardinals, it is reasonable to work with machines which model transfinite computations, i.e., computation processes that are allowed to run for transfinite time. This generalization of computable analysis to higher cardinalities was started by Galeotti and Nobrega in [19]. Their approach is based on Type 2 κ-Turing machines, the transfinite version of the Type 2 Turing machines originally introduced in [41]. In this section we introduce basic results of the theory of these generalized machines. We assume familiarity with basic notions of classical computability theory, and we refer the reader to [31] for more details.

1.2.1 Infinitary Turing Machines

We introduce the computational model of κ-Turing machines, which are the ordinal Turing machines introduced by Koepke in [25] with tapes of length κ and computations limited in time to κ (historically, Koepke's machines came after the invention of infinite time Turing machines by Hamkins and Lewis [22]). The idea of limiting the tape length of ordinal Turing machines to a fixed ordinal first appeared in Koepke's paper [26]. For a study on how the tape length influences computational power, we refer the reader to [32]. For more details about the theory of transfinite computability, we refer to [10].

Fix an infinite ordinal κ which is closed under ordinal multiplication. We define *κ-Turing machines*: the model comprises one input tape, an oracle tape, a parameter tape, an output tape and several scratch tapes, each divided in cells which can accommodate one single symbol $s \in \{0, 1\}$.[5] The cells on these tapes are indexed by ordinals in κ, and computations are allowed to run for $<\kappa$ steps. On each tape there is an independently movable head which can scan one cell at a time. The machine can modify the content of a given cell only when one of its heads is hovering over it. Its behaviour is determined by its configuration and its internal state.

Formally, the *configuration* of a κ-Turing machine with n tapes after α steps of computation can be expressed as a tuple $((T_i(\alpha))_{i\leq n}, (H_i(\alpha))_{i\leq n}, S(\alpha))$, where $T_i(\alpha)\colon \kappa \to 2$ indicates the content of the i-th tape, $H_i(\alpha) \in \kappa$ the position of the head on the i-th tape and $S(\alpha)$ indicates the internal state of the machine. We will sometimes refer to the read/write heads on the input and output tapes as H_{in} and H_{out}, respectively. Similarly we will refer to the input tape as T_{in} and the output tape as T_{out}. A *Turing program* P which mentions n tapes and internal states $Q = \{q_0, \ldots, q_n, q_{\text{lim}}\}$ is a finite set of instructions of the form

$$\mathtt{if}\ ((\mathtt{T_i})(\mathtt{H_i}))_{\mathtt{i\leq n}} = \mathtt{x}\ \mathtt{and}\ \mathtt{S} = \mathtt{q},\ \mathtt{write}\ \mathtt{y},\ \ \mathtt{move\ according\ to\ d\ and\ go\ into\ state}\ \mathtt{q}',$$

where $x, y \in \{0, 1\}^n$, $q, q' \in Q$ and $d \in \{\mathsf{L}, \mathsf{R}, \mathsf{S}\}^n$.

When a machine running program P finds itself in state q and in a configuration such that $T_i(H_i) = x(i)$ for all $i \leq n$, the machine executes the above instruction, replacing the content of the cells which are being scanned according to y (so $T_i(H_i) = y(i)$ for all $i \leq n$ at the next step of computation) and moves its heads according to d, so for all $i \leq n$, if $H_i = \alpha$ and $d(i) = \mathsf{R}$, H_i is moved to position $\alpha + 1$; if $d(i) = \mathsf{L}$, H_i is moved to position β if α is a successor ordinal and $\beta + 1 = \alpha$, and moved to position 0 if α is a limit ordinal; if $d(i) = \mathsf{S}$, H_i is not moved at all. The internal state of the machine is changed from q to q'.

The starting configuration of a κ-Turing Machine which uses n tapes, running on input $x \in 2^{<\kappa}$, with oracle $y \in 2^\kappa$ and parameter $\xi \in \kappa$ is the following: $S(0) = q_0$, $H_i(0) = 0$ for all $i \leq n$, the input tape is initialized as $T_1(0) = x0^\kappa$, the oracle tape is initialized as $T_2(0) = y$, the parameter tape is initialized as $T_3(0) = \chi_{\{\xi\}}$ and lastly $T_i(0) = 0^\kappa$ for all $4 \leq i \leq n$, so all other tapes are initially filled with zeros. Note that it is possible to allow the input x to be a bitstring of length κ without substantially modifying the description above and letting $T_1(0) = x$. Given an ordinal $\delta < \kappa$, the *(partial) computation up to step δ* of the machine M which is running program P, with input, oracle and parameter as described above is given by the sequence:

$$\big(((T_i(\alpha))_{i\leq n}, (H_i(\alpha))_{i\leq n}, S(\alpha)) \mid \alpha \in \delta\big),$$

where the configuration at time 0 is the one described above, for every successor ordinal $\alpha = \beta+1$, the configuration $((T_i(\alpha))_{i\leq n}, (H_i(\alpha))_{i\leq n}, S(\alpha))$ is determined by the configuration at time β by follwing the relevant instruction in P and for every limit ordinal λ, the configuration $((T_i(\lambda))_{i\leq n}, (H_i(\lambda))_{i\leq n}, S(\lambda))$ is determined by the so-called *limsup rule*, i.e.

$$H_i(\lambda) = \limsup_{\alpha\in\lambda} H_i(\alpha) \text{ for all } i \leq n,$$

and

$$T_i(\lambda)(\beta) = \limsup_{\alpha\in\lambda} T_i(\alpha)(\beta),$$

while the internal state at time λ is set to q_{lim}, a designated "limit state" which essentially gives us the freedom to assume that our machines know when they have performed limit-many steps of com-

[5]Other definitions allow every particular machine M to come with a finite alphabet Σ of possible symbols. It is well known that this difference is inconsequential for the theory (see, e.g., [10], Exercise 2.5.9). It is also known that any machine using finitely many tapes can be simulated by one using just one tape (see [10], Corollary 2.5.44). We prefer to explicitly mention multiple tapes as it makes the exposition clearer.

putation.[6] Notice that by induction we can prove that for any tape $i \leq n$ and any ordinal λ, $H_i(\lambda) \leq \lambda$.

We will generally assume that our machines have a way to know when they have finished reading their input, and similarly that they can "signal" the end of their output. This can be achieved by considering machines using symbols in $\{0, 1, \mathsf{E}\}$ and assuming that, whenever we speak of some $x \in 2^{<\kappa}$ being the input/output of a κ-Turing Machine, what we mean is that $x\mathsf{E}$ is what actually appears on the tape. Similarly we can make sure that our machines can recognize the beginning of the tape. Another useful fact about κ-Turing Machines is that we can safely assume that any machine M is always "aware" of the position of its heads. This is achieved as follows: given a program P which mentions n tapes, consider the program P' which mentions $2n$ tapes and, besides simulating P, moves its heads on tapes $T_{n+1}, \ldots, T_{2n}$ in a way to have $H_i(\alpha) = H_{n+i}(\alpha)$ for all ordinals α. Now say that we want to know the value of H_i at a certain step of computation: we simply need to write the symbol 1 on the current position on T_{n+i} and move H_{n+i} to the starting cell (note that this can be achieved in a finite number of steps by simply going left indefinitely until the head "falls" from a limit position to cell 0). At this point, tape T_{n+i} contains the representation of a certain ordinal β which is exactly the position of H_i, so we can, e.g., perform any computable ordinal operation with it (cf. the proof of Proposition 2.5.3 in [10]). In the rest of this thesis, we will use of these facts several times, without explicitly mentioning them.

A κ-Turing machine M running the program P *halts* at step $\delta < \kappa$ if its computation can be defined up to and including the ordinal δ but M is unable to apply any of the instructions in P to its configuration at time δ, so the configuration at time $\delta + 1$ cannot be defined. In this case, we call δ the *halting time* of the computation, and we say that $T_{\text{out}}(\delta)$ is the *output* of the computation (more precisely, we will often say that the output of the computation is $T_{\text{out}}(\delta)\upharpoonright\lambda$, where $\lambda = H_{\text{out}}(\delta)$). In this case the sequence of configurations of M up to and including step δ is the entire computation of M. If this does not happen, then we say that M *computes indefinitely* (or *diverges*) on the given input and we consider the sequence $\big(((T_i(\alpha))_{i\leq n}, (H_i(\alpha))_{i\leq n}, S(\alpha)) \mid \alpha \in \kappa\big)$ as the computation of M.

From now on we will, unless explicitly mentioned, always consider Turing programs that use their output tape in an "append only" way, in other words, the head of the output tape is only ever moved to the right. This means that if a symbol is written on the cell indexed with α on the output tape, that symbol is not erased in the following steps of computation (note that this is compatible with the lim sup rule for configurations at a limit ordinal time λ as the content of any cell on the output tape is constant cofinally in λ). It is clear that this condition can be achieved by limiting the set of admissible instructions to those which do not include moving the output head to the left. Moreover, it is obvious that this difference is inconsequential in the sense that for every Turing program P there exists a program P' which uses its output tape in an "append only" fashion and such that for every choice of parameter ξ, oracle y and input x, P halts if and only if P' halts, and their outputs are the same.
Note that the Turing programs which can be run on κ-Turing Machines are finite sets of instructions and they are exactly the same as the ordinary ones, hence there exists an effective enumeration $(P_i)_{i\in\omega}$ of every possible Turing program.

We use terminology from computability theory in the expected way: in particular we say that (partial) function $f :\subseteq 2^{<\kappa} \to 2^{<\kappa}$ is *κ-computable* if and only if there exists a Turing program P and a parameter δ such that a machine M running program P with parameter δ halts on input $w \in 2^{<\kappa}$ iff $w \in \text{dom}(f)$ and if this is the case, it outputs $f(\beta)$; a set $A \subseteq 2^{<\kappa}$ is *κ-recursively enumerable* (or *κ-semidecidable*) if it is the domain of a computable partial function; and lastly that A is *κ-recursive* (or *κ-decidable*, *κ-computable*) if both A and $2^{<\kappa} \setminus A$ are recursively enumerable. Similarly we can consider computability on κ as follows: we say that a machine M takes as input an ordinal $\delta < \kappa$

[6] We mention that this convention is adopted for ease of notation and it is not actually an extra assumption on the computational power of κ-Turing Machines: it is easy to see that using so-called flag bits is sufficient for our machines to be aware of limit times (see [10], Remark 2.2.7).

if $T_{\text{in}}(0) = \chi_{\{\delta\}}$ and that it yields as output $\beta < \kappa$ if M halts on input δ and, if λ is the halting time, $T_{\text{out}}(\lambda) = \chi_{\{\beta\}}$. In these terms, a (partial) function $f :\subseteq \kappa \to \kappa$ is *κ-computable* if and only if the corresponding function on $2^{<\kappa}$ is. The notions of recursive enumerability and recursiveness for subsets of κ are defined as above. Finally, for any given $i, j, \ell, k \in \omega$, we can consider computability of functions $(2^{<\kappa})^i \times \kappa^j \to (2^{<\kappa})^\ell \times \kappa^k$ by considering machines with multiple input/output tapes.

In analogy with classical computability theory, we say that a set $A \subseteq \kappa$ is *κ-computable realtive to* $B \subseteq \kappa$ if χ_A is κ-computable using χ_B as an oracle. In this case we also say that A is *κ-Turing reducible* to B and we denote this as $A \leq^\kappa_{\text{T}} B$. We extend these notions to functions in κ^κ in the same way as it is done in the classical setting. It is clear that $\leq^\kappa_{\text{T}}$ is a preorder, hence the relation $\equiv^\kappa_{\text{T}}$ defined as $A \equiv^\kappa_{\text{T}} B$ if and only if $A \leq^\kappa_{\text{T}} B$ and $B \leq^\kappa_{\text{T}} A$ is an equivalence relation. We will call equivalence classes of this relation *κ-Turing degrees.* We use the symbol $\oplus$ to indicate the Turing join on κ-Turing degrees, defined in the same way as it is in the classical setting, see, e.g., [31, Proposition V 1.2]. When there is no risk of confusion, we will drop the κ- prefix and simply use the terms computable, recursive, decidable, Turing degrees and the symbol $\leq_{\text{T}}$ to refer to the above notions.

In the following we will always consider computations using parameters: it's easy to see that, in contrast with ordinary Turing machines, allowing parameters actually makes a difference. This is explained by the fact that if κ is an uncountable ordinal, then there exist ordinals $\delta < \kappa$ which are not computable by a κ-Turing Machine using no parameter running on the empty input (this follows by a simple cardinality argument and it is in contrast with the classical setting, as every natural number is obviously computable by some Turing machine running on the empty input). If P is a Turing program and $\delta < \kappa$ an ordinal, we call the pair (P, δ) a program-parameter pair and we denote by T the set of all such pairs. In the rest of this thesis we will often use the term "(κ-)Turing machine" to refer to a program-parameter pair. It is clear that there are $\aleph_0$ Turing programs and κ possible parameters, for a total of κ possible Turing machines, and that given an injective pairing function $\omega \times \kappa \to \kappa$, we can exploit any of the classical effective enumerations of Turing programs to obtain an enumeration of all possible program-parameter pairs (and consequently of all possible partial computable functions $2^{<\kappa} \to 2^{<\kappa}/\kappa \to \kappa$, of all recursively enumerable subsets of $2^{<\kappa}/\kappa$, etc.).

We close this introductory section with a result listing various basic operations on ordinals which turn out to be computable.

Lemma 1.2.1. *The following relations are computable by κ-Turing machines: every Turing computable relation, comparisons of ordinals , ordinal addition and ordinal multiplication restricted to ordinals $<\kappa$.*

Proof. The fact that Turing computable relations are computable by κ-Turing Machines follows from the fact that the latter can simulate ordinary Turing Machines, for the operations on ordinals see Section 3 "Ordinal algorithms" in [25]. $\square$

1.2.2 Pairing functions

In this section we fix some notation for well-known pairing functions and we set some conventions on symbols which will have a precise meaning for the rest of this thesis. All the pairing functions we introduce are well known tools to work on sequences of ordinals with Turing machines. Most of these functions are ultimately based on Gödel's pairing function on ordinals.

Convention 1.2.2. *In the rest of this thesis κ is a fixed uncountable cardinal satisfying $\kappa^{<\kappa} = \kappa$.*

Note that Kőnig's theorem implies that $\kappa^{\text{cf}(\kappa)} > \kappa$, hence the assumption above implies in particular that κ is regular.
Define the following order on $\kappa \times \kappa$:

$$(\alpha, \beta) \prec (\gamma, \delta) \iff \begin{cases} \max\{\alpha, \beta\} < \max\{\gamma, \delta\} \text{ or} \\ \max\{\alpha, \beta\} = \max\{\gamma, \delta\} \wedge \alpha < \gamma \text{ or} \\ \max\{\alpha, \beta\} = \max\{\gamma, \delta\} \wedge \alpha = \gamma \wedge \beta < \delta \end{cases}$$

It is known that this is a well ordering of $\kappa \times \kappa$ of type κ. We can use $\prec$ to define the *pairing function* $\ulcorner \cdot, \cdot \urcorner \colon \kappa \times \kappa \to \kappa$ as

$$\ulcorner \alpha, \beta \urcorner = \mathrm{ot}\{(\gamma, \delta) \mid (\gamma, \delta) \prec (\alpha, \beta)\}.$$

The fact that $\prec$ is a well-order of type κ of the set $\kappa \times \kappa$ immediately implies that the function $\ulcorner \cdot \urcorner$ is a bijection. We will denote its inverse as $\mathrm{unpair} \colon \kappa \to \kappa \times \kappa$ and we will sometimes refer to it as the *unpairing function*. The following is a fundamental, well known result (see e.g., [19]) and will be useful throughout the thesis:

Lemma 1.2.3. *The pairing function* $\ulcorner \cdot \urcorner$ *and its inverse* unpair *are κ-computable.*

Proof. See [30], Lemma 4.25 or [10], Lemma 2.3.37 and Lemma 2.3.38. □

This immediately leads to the possibility of making our enumeration of the set of κ-Turing machines T effective.

Corollary 1.2.4. *There exists an effective enumeration of the set T of all κ-Turing Machines.*

From this, a straightforward generalization of the classical proof (see, e.g., pg. 22 in [33]) yields:

Corollary 1.2.5. *There exists a universal κ-Turing Machine and κ-Turing Machines are closed under recursion and composition.*

We exploit the pairing function to define the *interleaving function* $\langle \cdot \rangle \colon (\kappa^\kappa)^\kappa \to \kappa^\kappa$, which makes a κ-sequence of κ-long words into a single κ-long word as follows

$$\langle (p_\alpha)_{\alpha \in \kappa} \rangle (\gamma) = p_\alpha(\beta)$$

if and only if $\gamma = \ulcorner \alpha, \beta \urcorner$. For a fixed $\alpha \in \kappa$ we define $\cdot_\alpha \colon \kappa^\kappa \to \kappa^\kappa$ as

$$p_\alpha(\beta) = p(\ulcorner \alpha, \beta \urcorner).$$

It is immediate to see that for every $p \in \kappa^\kappa$:

$$p = \langle (p_\alpha)_{\alpha \in \kappa} \rangle.$$

Define $\langle \cdot, \cdot \rangle \colon \kappa^\kappa \times \kappa \to \kappa^\kappa$ as

$$\langle p, \alpha \rangle = \alpha ^\frown p,$$

and inductively define $\langle \cdot, \ldots, \cdot \rangle_n \colon \kappa^\kappa \times \kappa^n \to \kappa^\kappa$ as

$$\langle p, \alpha_0, \ldots, \alpha_{n-1} \rangle_n = \langle \langle p, \alpha_0, \ldots, \alpha_{n-2} \rangle_{n-1}, \alpha_{n-1} \rangle.$$

Moreover, for any $n \in \omega$, define $\langle \cdot, \ldots, \cdot \rangle_n \colon (\kappa^\kappa)^n \to \kappa^\kappa$ as

$$\langle p_1, \ldots, p_n \rangle_n(\alpha) = p_i(\gamma)$$

if and only if α is the γ-th ordinal congruent to i modulo n. In the rest of this thesis we will usually omit the subscripts from the symbols for these functions. Using the same symbols for different functions will lead to no ambiguity as the intended function is always understood from the context. Lastly we introduce tupling functions to handle concatenations of words of length $<\kappa$ in a way that is suitable for κ-Turing machines: given $\lambda < \kappa$ and $a \in \kappa^\lambda$, define

$$\iota(a) = 11[\![00^{a(\beta)}0]\!]11.$$

Now if $(w_\alpha)_{\alpha \in \lambda}$ is a sequence of words in $\kappa^{<\kappa}$, $p \in \kappa^\kappa$ and $\lambda \leq \kappa$, define

$$\langle w_1, p \rangle = \langle p, w_1 \rangle = \iota(w_1)p,$$

and

$$\langle (w_\alpha)_{\alpha \in \lambda} \rangle = [\![\iota(w_\alpha)]\!]_{\alpha \in \lambda}.$$

Note that this allows us to define a notion of computability on $\kappa^{<\kappa}$ as follows: we say that $f :\subseteq \kappa^{<\kappa} \to \kappa^{<\kappa}$ is computable if and only if $\iota \circ f \circ \iota^{-1}$ is. The notions of recursive enumerability and recursiveness for subsets of $\kappa^{<\kappa}$ are then defined as usual.

1.2.3 Enumeration of short sequences

Many algorithms in the rest of this thesis will need to enumerate the sets $2^{<\kappa}$ and $\kappa^{<\kappa}$. In this section, we shall show that the existence of a κ-computable enumeration of $2^{<\kappa}$ is equivalent to the axiom of constructibility at κ (i.e., $\mathrm{V}_\kappa = \mathrm{L}_\kappa$). The fact that $\mathrm{V}_\kappa = \mathrm{L}_\kappa$ implies the existence of a κ-computable enumeration of $2^{<\kappa}$ is due to Koepke and its proof makes use of his bounded truth predicate for L. These tools can be found in his papers [26] and [25]. The opposite implication is obtained as a direct consequence of the results in Ethan Lewis' masters thesis [27]. In this section we assume basic facts about the constructible universe L. For the proof of these facts and more details on L we refer the reader to [15].

Lemma 1.2.6. *There exists a κ-computable surjective function $f\colon \kappa \to \kappa^{<\omega}$.*

Proof. See Section 4 in [25], where the author builds an OTM computable enumeration of all finite sets of ordinals, and note that the restriction of this function to κ is κ-computable and it is a surjection on $\kappa^{<\omega}$. □

Now we define so-called *bounded formulas* and *bounded terms* as follows:

(a) the variable symbols $\{v_i \mid i \in \omega\}$ are bounded terms,

(b) if s, t are bounded terms, then $s \in t$ and $s = t$ are bounded formulas,

(c) if φ and ψ are bounded formulas, $i, j \in \omega$, then $\varphi \wedge \psi$, $\neg\varphi$, $\exists v_i \in v_j \varphi(v_i)$, $\forall v_i \in v_j \varphi(v_i)$ are bounded formulas,

(d) if φ is a bounded formula, $i, j \in \omega$, then $\{v_i \in v_j \mid \varphi(v_i)\}$ is a bounded term,

We restrict attention to so-called *tidy* bounded formulas, i.e.,those in which every free variable occurs exactly once and no bound variable occurs free. Call F the set of all such formulas: by arithmetization of sintax, it is clear that F can be effectively enumerated by a Turing machine, and consequently it can be enumerated by a κ-Turing Machines.

Given an assignment function $a\colon n \to \mathrm{V}$, we say that $a(i)$ is the interpretation of v_i under a and if t is a bounded term we write $t[a]$ to denote the interpretation of t under assignment a.

Define the class $A = \mathrm{Ord}^{<\omega} \times F$ and define $W\colon A \to 2$ as:

$$W((\gamma, \gamma_0, \dots, \gamma_n), \varphi) = 1 \iff \mathrm{L}_\gamma \vDash \varphi(\mathrm{L}_{\gamma_0}, \dots, \mathrm{L}_{\gamma_n})$$

It is the central result in [25] that W is computable by so-called ordinal Turing machines (OTMs), which are machines analogous to the κ-Turing Machines introduced above, with the only difference being that they have no limitation on tape size and computation length. We note that any OTM computation with input length $\gamma < \kappa$ and halting time $\theta < \kappa$ can be carried out on a κ-Turing Machine. We state a refinement of this result, also due to Koepke:

Lemma 1.2.7. *There is a Turing program P_{truth} that κ-computes the restriction of W to $\kappa^{<\omega} \times F$.*

Proof. See [26], Lemma 5. □

It can be shown that every set x in L is ordinal definable, meaning that in particular there exists a tidy bounded term t and ordinals $\alpha_0, \dots, \alpha_n$ such that $t[(\alpha_0, \dots, \alpha_n)] = x$. Combining this fact with the computability of W one can show:

Lemma 1.2.8. *For every constructible $x \subseteq \mathrm{Ord}$, there is a Turing program P and ordinals $\alpha_0, \dots, \alpha_n$ such that running P (on an OTM) with input $\chi_{\{\alpha_0, \dots, \alpha_n\}}$ leads to a terminating computation with output χ_x.*

Proof. See [25], Theorem 6.2. □

Note that the program P mentioned in the statement of Lemma 1.2.8 is nothing more than a Turing program which checks whether $x = t[(\alpha_0, \ldots, \alpha_n)]$ by means of the truth function W. By Lemma 1.2.7, as long as the parameters $\alpha_0, \ldots, \alpha_n$ are below κ, the program P can be run on a κ-Turing Machine with the same results.

Exploiting an absoluteness result on ordinal computation, one can show that

Lemma 1.2.9. *The constructible universe* $(\mathrm{L}, \in)$ *proves that every set of ordinals is ordinal computable from a finite set of ordinals. Conversely, every ordinal computable set of ordinals is constructible.*

Proof. See [25], Theorem 7.1 and Theorem 6.2. □

The first part of this result relativizes to κ, i.e.,if $\mathrm{V}_\kappa = \mathrm{L}_\kappa$, then every subset of κ is ordinal computable from a finite set of ordinals.

Now we can state the main result of this section, which immediately implies the existence of an effective enumeration of $2^{<\kappa}$:

Theorem 1.2.10. *Assume that any bounded subset of* κ *is ordinal computable from a finite set of ordinals, let* $\omega \leq \lambda < \kappa$ *and let* $x \subseteq \lambda$*, then there exists a Turing program* P *and a finite set of ordinals* $\alpha_0, \ldots, \alpha_n$*, with* $\alpha_i < |\lambda|^+$ *such that running* P *(on an OTM) with input* $\chi_{\{\alpha_0, \ldots, \alpha_n\}}$ *results in a terminating computation with output* χ_x*. Moreover,* P *halts in* $<|\lambda|^+$ *steps.*

Proof. See [25], Theorem 7.2(a). □

This final result immediately implies that for every $x \in \mathrm{L}$, if $x \subseteq \lambda$ for some $\lambda < \kappa$, then there are parameters $\alpha_0, \ldots, \alpha_n < \kappa$ such that χ_x is computable by a κ-Turing Machine from $\alpha_0, \ldots, \alpha_n$. Note that if $x \in \mathrm{L}$ and $x \subseteq \lambda$, then $x \in \mathrm{L}_{|\lambda|^+} \subseteq \mathrm{L}_\kappa$.

Corollary 1.2.11. *Assume* $\mathrm{V}_\kappa = \mathrm{L}_\kappa$*, then there exists a Turing machine* M *which* κ*-computes a total surjective function* $\nu' \colon \kappa \to 2^{<\kappa}$*.*

Proof. On input $\alpha \in \kappa$, start running through all ordinals $\delta \in \kappa$. For each of these (in parallel), use the unpairing function to determine whether it codes an element of $A \cap \kappa^{<\omega} \times T$. If it doesn't, go to the next ordinal. If it does, extract the pair (a, P) and simulate P on input χ_a. Update a counter to know how many of these simulating computation have halted. Clearly there are κ pairs which lead to a halting computation, so at some point the machine finds the α-th pair $(a', P') \in \kappa^{<\omega} \times T$ such that P' halts on input $\chi_{a'}$. At this point, print the output of the simulated computation and terminate. By Theorem 1.2.10, for every $\lambda < \kappa$ and every $x \subseteq \lambda$, there is a pair (a'', P'') which outputs χ_x, hence there must be some β such that M outputs χ_x when run on input β. This shows that the function f computed by M is surjective. The fact that $\mathrm{dom}(f) = \kappa$ is clear from its definition. □

Inspection of this proof immediately yields a "padding lemma" for ν':

Lemma 1.2.12. *For every* $w \in 2^{<\kappa}$ *and every* $\beta \in \kappa$ *there exists some* $\gamma > \beta$ *such that* $\nu'(\gamma) = w$*.*

Proof. Let $w \in 2^{<\kappa}$, $\beta \in \kappa$ be as above and let $\delta \in \kappa$ be such that $w = \nu(\delta)$, so δ codes a pair $(a, P) \in A \cap \kappa^{<\omega} \times T$ such that running P on χ_a results in a halting computation with output w. Consider $\delta' \in \kappa$ such that δ' codes (a', P') with $a' = \beta^\frown a$ and P' being the same program-parameter pair as P, with the only difference being that P' has an additional input tape (which is used to store β), but completely ignores said tape. Clearly $\delta' > \beta$ and clearly the output of P' when run on a' is w. □

From this, we can refine the statement of Corollary 1.2.11

Corollary 1.2.13. *Assume* $\mathrm{V}_\kappa = \mathrm{L}_\kappa$*, then there exists a* κ*-computable total surjective function* $\nu \colon \kappa \to 2^{<\kappa}$ *such that for every* $w \in 2^{<\kappa}$ *if* $\mathrm{dom}(w) = \alpha$*, then for all* $\beta \in \kappa$*,* $\nu(\beta) = w$ *implies* $\beta > \alpha$*.*

Proof. Consider the function ν' and modify it in the following way: given β, compute $\nu'(\beta)$. If $|\nu'(\beta)| \geq \beta$, output the empty sequence and terminate. Otherwise, output $\nu'(\beta)$. This is a computable procedure, hence the function ν it defines is κ-computable. Surjectivity of ν follows at once from Lemma 1.2.12. □

In the rest of this thesis we will always use the function ν because the property $|\nu(\beta)| < \beta$ is arguably a natural one and it will be exploited in some of our results. Note that the analogous property holds for all reasonable effective enumerations of $2^{<\omega}$.
Lastly, consider the funciton $\iota\colon \kappa^{<\kappa} \to 2^{<\kappa}$ introduced in the previous section: it is clear that this function is an injection, so we can interpret each $p \in \mathrm{ran}(\iota)$ as a code for a unique sequence in $\kappa^{<\kappa}$. Moreover, $\mathrm{ran}(\iota)$ is a decidable subset of $2^{<\kappa}$, hence the machine computing ν can be easily modified to yield:

Corollary 1.2.14. *Assume $\mathrm{V}_\kappa = \mathrm{L}_\kappa$, then there exists a κ-computable surjection $\mu\colon \kappa \to \kappa^{<\kappa}$.*

Note that by our definition for every $w \in \kappa^{<\kappa}$, $\mu(\beta) = w$ iff $\nu(\beta) = \iota(w)$, and since $|\iota(w)| \leq \beta$ we obtain $|w| \leq \beta$. This means that μ also has the property that for all $\lambda \in \kappa$, no word of length λ is in $\mathrm{ran}(\mu\restriction(\lambda+1))$.

Remark 1.2.15. *We introduced κ-Turing Machines as pairs given by a Turing program P together with an ordinal $\delta < \kappa$ which performs the function of a parameter. The fact that the functions μ and ν allow us to store any $p \in (2^{<\kappa} \cup \kappa^{<\kappa})$ in a single ordinal makes many operations with such objects computable from their code, e.g., given any $\lambda \in \kappa$ and any code for a sequence $w \in \kappa^\lambda$, problems such as determining whether a given δ belongs to $\mathrm{ran}(w)$ or whether a given $w' \in \kappa^{<\kappa}$ is an initial segment of w are decidable.*

We conclude the section by showing that the assumption $\mathrm{V}_\kappa = \mathrm{L}_\kappa$ is optimal if we want to computably enumerate $2^{<\kappa}$. This is a known fact, and it is the relativization to κ of [27, Theorem 5.11].

Proposition 1.2.16. *If $2^{<\kappa} \subseteq \mathrm{L}$, then $\mathrm{V}_\kappa = \mathrm{L}_\kappa$.*

Proof. We only have to show that $\mathrm{V}_\kappa \subseteq \mathrm{L}_\kappa$ as the reverse inclusion always holds. By the relativization to κ of [27, Lemma 5.8] we obtain that the restriction of the L well-ordering $<_{\mathrm{L}}$ to $2^{<\kappa}$ can be used to define a well-order $\prec$ of V_κ of type κ such that for any two sets $x, y \in \mathrm{V}_\kappa$, if $x \in y$, then $x \prec y$. For any set $x \in \mathrm{V}_\kappa$, denote as α_x the order type of the set $\{z \in \mathrm{V}_\kappa \mid z \prec x\}$. Now, by contradiction assume $x \in \mathrm{V}_\kappa \setminus \mathrm{L}_\kappa$ and by $\in$-induction assume that x is of least rank among sets in $\mathrm{V}_\kappa \setminus \mathrm{L}_\kappa$. We obtain that the string $\chi_x\colon \alpha_x \to 2$ given by $\chi_x(\alpha) = 1$ if and only if $\alpha = \alpha_y$ for some $y \in x$ is a complete description of x, i.e.,$y \in x$ if and only if $\alpha_y \in \mathrm{dom}(\chi_x)$ and $\chi_x(\alpha_y) = 1$. Furthermore, this implies that $|x| \leq |\alpha_x| < \kappa$ and, since every element of x is in L_κ and κ is regular, we obtain that $x \subseteq \mathrm{L}_\eta$ for some $\eta < \kappa$. Since $\chi_x \in 2^{<\kappa}$, by assumption we have $\chi_x \in \mathrm{L}_\kappa$ and therefore there must be some $\beta < \kappa$ such that $\chi_x \in \mathrm{L}_\beta$ and $x \subseteq \mathrm{L}_\beta$. We obtain that $x = \{y \in \mathrm{L}_\beta \mid \chi_x(\alpha_y) = 1\}$, therefore $x \in \mathrm{L}_{\beta+1} \subseteq \mathrm{L}_\kappa$. This is the required contradiction, so it must be the case that $\mathrm{V}_\kappa = \mathrm{L}_\kappa$. □

Since every computable string must be constructible, if there is a κ-computable enumeration of $2^{<\kappa}$, then $2^{<\kappa} \subseteq \mathrm{L}$. Hence we obtain:

Corollary 1.2.17. *If there exists a computable enumeration of $2^{<\kappa}$, then $\mathrm{V}_\kappa = \mathrm{L}_\kappa$.*

This concludes the proof of the equivalence between these two assumptions. Note that for every ordinal α, we know that $|\,\mathrm{V}_{\omega+\alpha}\,| = \beth_\alpha$ (cf. [23, Exercise 6.2]) whereas for any infinite ordinal β, $|\,\mathrm{L}_\beta\,| = |\beta|$ (cf. [23, Exercise 13.19]). So, we obtain that the assumption of κ-enumerability of $2^{<\kappa}$ in particular implies that $\kappa = \beth_\kappa$.

Remark 1.2.18. *In the rest of this thesis, whenever we write ν, we mean an arbitrary enumeration of $2^{<\kappa}$ satisfying the length bound $\mathrm{dom}(\nu(\alpha)) < \alpha$ for all $\alpha < \kappa$. Similarly, we use μ to refer to any enumeration of $\kappa^{<\kappa}$ satisfying the length bound $\mathrm{dom}(\mu(\alpha)) < \alpha$ for all $\alpha < \kappa$.*

We signal the results which require the assumption of the computability of these enumerating functions. We state this assumption here for future reference.

Hypothesis 1.2.19. *The functions $\nu\colon \kappa \to 2^{<\kappa}$ and $\mu\colon \kappa \to \kappa^{<\kappa}$ are computable, or equivalently* $\mathrm{V}_\kappa = \mathrm{L}_\kappa$.

Note that, in this assumption, both ν and μ have computable right inverses: given $w \in \lambda^{<\kappa}$ where $\lambda \in \{2, \kappa\}$, we can run through all ordinals below κ and compute the corresponding word until we find the first ordinal δ which gives output w and output it.[7]

1.2.4 Type 2 computations

The field of classical computable analysis makes extensive use of so-called Type 2 computations, which give a good notion of computability for functions $f :\subseteq 2^\omega \to 2^\omega$. This notion of computability is of central importance as it allows us to define a notion of computability for spaces of cardinality up to $2^{\aleph_0}$. In [19] the authors introduced Type 2 computations in the context of κ-Turing Machines and consequently defined a notion of computability for functions on generalized Cantor space 2^κ. We give the relevant definitions and prove some basic facts about Type 2 κ-computability.

Definition 1.2.20 (Type 2 computability)**.** *Let $f :\subseteq 2^\kappa \to 2^\kappa$ be a partial function and M a κ-Turing machine, we say that M* Type 2 computes *f if and only if for all $x \in \mathrm{dom}(f)$, M computes indefinitely on input x, for every $\beta < \kappa$ there exists $\gamma < \kappa$ such that $H_{\text{out}}(\gamma) = \beta$ and $T_{\text{out}}(\gamma)\restriction\beta \subseteq f(x)$.*

Accordingly, $f :\subseteq 2^\kappa \to 2^\kappa$ is said to be *Type* 2 *computable* if there exists a Turing machine which Type 2 computes it. Notice that although Type 2 computability is defined by the same Turing machines as regular computability, we will at times refer to machines computing functions on generalized Cantor space as Type 2 κ-Turing machines (T2κTMs for short). In the rest of this thesis we will often refer to ordinary computability as Type 1 computability.

We can export the notion of computability on 2^κ obtained with T2κTMs to a notion of computability on κ^κ as follows: consider the function $i\colon \kappa^\kappa \to 2^\kappa$ given by

$$i(p) = [\![1100^{p(\alpha)}011]\!]_{\alpha\in\kappa}.$$

It is clear that i is an injection, call $\delta'_{\kappa^\kappa} :\subseteq 2^\kappa \to \kappa^\kappa$ its left inverse with minimal domain,[8] and for every $p \in \kappa^\kappa$, call $i(p)$ *the code* for p.

We stipulate that a function $f :\subseteq \kappa^\kappa \to \kappa^\kappa$ is Type 2 computable if and only if *the unique* function $f' :\subseteq 2^\kappa \to 2^\kappa$ with domain $i[\mathrm{dom}(f)]$ such that for every $p \in \mathrm{dom}(f)$, $f'(i(p)) = i(f(p))$ is Type 2 computable. In the rest of this thesis, we will use the notions of Type 2 computability on 2^κ and on κ^κ interchangeably, in particular, most of the representations introduced in the later chapters and most of the algorithms used will refer to κ^κ rather than 2^κ. Whenever we will argue for the existence of a given κ-Turing Machine which Type 2 computes functions on κ^κ, it will be clear that such a machine can actually be realized as a machine operating with our standard $\{0, 1\}$ alphabet.

Proposition 1.2.21. *Let $F :\subseteq 2^\kappa \to 2^\kappa$ be any function, then F is Type* 2 *computable if and only if there exists a Type* 1 *computable $f :\subseteq 2^{<\kappa} \to 2^{<\kappa}$ such that*

$$A = \{w \in 2^{<\kappa} \mid \exists p \in \mathrm{dom}(F)\ w \subseteq p\} \subseteq \mathrm{dom}(f),$$

for $w, w' \in A$

$$w \subseteq w' \implies f(w) \subseteq f(w'),$$

and lastly for all $x \in \mathrm{dom}(F)$

$$F(x) = \bigcup_{\alpha\in\kappa} f(x\restriction\alpha).$$

The same holds relative to any oracle $o \in 2^\kappa$.

[7]It is clear from our proof sketch that this is an instance of a more general principle: if f is a computable function with $\mathrm{dom}(f)$ computably enumerable, then f has a computable right inverse.

[8]The partial function δ'_{κ^κ} is an example of the representations which we will introduce in section 1.3. Note that δ'_{κ^κ} is an injection.

Proof. It is clear that if F is Type 2 computable we can modify the corresponding machine M to obtain a machine M' computing an appropriate f. Conversely, if there exists a computable f as above, we can obtain a machine computing F as follows: given $p \in 2^\kappa$ as input, we run the machine for f in parallel on $p{\restriction}\alpha$ for all $\alpha \in \kappa$ and we update the outputs with the new bits as they come. It is obvious that the proof relativizes to any oracle. $\square$

Similarly to the classical case, there is a close relation between Type 2-computable functions and continuous functions on Cantor space. We state this result, originally from [19].

Proposition 1.2.22. *Let $F :\subseteq 2^\kappa \to 2^\kappa$ be a partial function, then F is continuous if and only if F is Type 2 κ-computable with respect to some oracle $o \in 2^\kappa$.*

Proof. See [30] Theorem 4.24. $\square$

It is straightforward to see that the same also holds for partial functions on κ^κ.

Proposition 1.2.23. *Let $p, o \in \kappa^\kappa$, then p is a total Type 1 computable function relative to o if and only if p is Type 2 computable with input o, i.e.,there exists some T2κTM which prints p on input o.*

Proof. Assume p is computable in the oracle o, then there exists a κ-Turing Machine M which, on input $\alpha \in \kappa$, outputs $p(\alpha)$. Modify M as follows: run through all ordinals $\alpha \in \kappa$ and for each of them in succession print $p(\alpha)$ on the output tape. This results in a κ-long computation of a machine M' which uses o as input and in the long run prints p on the output tape. Conversely assume p is Type 2 computable from input $o \in \kappa^\kappa$ with T2κTM M. Modify M as follows: given an ordinal α and oracle access to o, simulate the computation of M up to the α-th bit of output, then output $p(\alpha)$. This results in a machine M' which computes the function p with oracle access to o. $\square$

Note that, as can be seen in the proof above, the relation between inputs and oracles in Type 2 computations is close to the relation between inputs and parameters in Type 1 computations. Moreover, we derive the following corollary:

Corollary 1.2.24. *Let $p \in \kappa^\kappa$ be computable relative to oracle $o \in \kappa^\kappa$ and let $G :\subseteq \kappa^\kappa \to \kappa^\kappa$ be a Type 2 κ-computable function with $p \in \mathrm{dom}(G)$, then $G(p)$ is computable with respect to the oracle o. In particular, Type 2 computable functions can only decrease the Turing degree of their inputs.*

Proof. The assumption that p is computable with oracle o means that there exists a T2κTM M such that $M(o) = p$. Composing such machine with the machine computing G yields a T2κTM which computes $G(p)$ with input o, i.e.,$G(p)$ is computable relative to o. $\square$

In light of Proposition 1.2.23, we know that no confusion can arise when we refer to an element of κ^κ as computable (relative to an oracle), and we will use either characterization depending on context.

We state and prove a result which is particular of κ-Turing machines, as it exploits the property $\mathrm{cf}(\kappa) > \omega$. Given a Type 2 computation with input x, we can define a function ℓ_x which intuitively keeps track of the speed of the computation.

Definition 1.2.25 (Time function)**.** *Let $f :\subseteq 2^\kappa \to 2^\kappa$ be a T2κ-computable function, and let M be the Turing machine computing it. For any $x \in \mathrm{dom}(f)$, define the function $\ell_x\colon \kappa \to \kappa$ as*

$$\ell_x(\beta) = \min\{\delta \in \kappa \mid H_{\mathrm{out}}(\delta) = \beta\}.$$

We show that such functions are *normal*:

Lemma 1.2.26. *Let $\ell_x\colon \kappa \to \kappa$ be as in Definition 1.2.25, then ℓ_x is strictly increasing and continuous, i.e.,for every limit ordinal $\lambda < \kappa$,*

$$\ell_x(\lambda) = \sup_{\alpha \in \lambda} \ell_x(\alpha).$$

Proof. The fact that ℓ_x is strictly increasing is immediate by definition. To show that it is continuous, let $\delta < \kappa$ be a limit ordinal and define

$$\gamma = \sup\{\ell_x(\beta) \mid \beta \in \delta\}.$$

It is clear that γ is a limit ordinal, therefore

$$H_{\text{out}}(\gamma) = \limsup_{\alpha<\gamma} H_{\text{out}}(\alpha).$$

Since H_{out} is a nondecreasing function, we obtain

$$H_{\text{out}}(\gamma) = \sup_{\alpha<\gamma} H_{\text{out}}(\alpha).$$

Now the sequence $(\ell_x(\beta))_{\beta\in\delta}$ is an increasing sequence cofinal in γ and H_{out} is a nondecreasing function, therefore $\{H_{\text{out}}(\ell_x(\beta)) \mid \beta \in \delta\}$ is cofinal in $\{H_{\text{out}}(\alpha) \mid \alpha \in \gamma\}$. By definition we have $H_{\text{out}}(\ell_x(\beta)) = \beta$ for all $\beta \in \delta$, hence we can combine the above results to obtain that δ is cofinal in $\{H_{\text{out}}(\alpha) \mid \alpha \in \gamma\}$. From this it follows that the two sets have the same supremum, in other words, $H_{\text{out}}(\gamma) = \delta$. By definition this implies that $\ell_x(\delta) \leq \gamma$. We prove by contradiction that actually $\ell_x(\delta) = \gamma$. Assume $\ell_x(\delta) = \alpha < \gamma$, so α is the first time step at which H_{out} hovers over the cell indexed by δ. By definition of γ, there exists some $\beta < \delta$ such that $\ell_x(\beta) = \epsilon > \alpha$, i.e.,the first time step at which H_{out} hovers over the cell indexed by β is strictly greater than the first time step at which H_{out} hovers over the cell indexed by δ. This is a contradiction, hence

$$\ell_x(\delta) = \gamma = \sup\{\ell_x(\beta) \mid \beta \in \delta\},$$

which proves continuity. □

This immediately yields:

Lemma 1.2.27 (Fixpoint lemma for the time function)**.** *Let ℓ_x be as in definition 1.2.25 and let $\alpha < \kappa$. There exists $\beta \geq \alpha$ such that $\ell_x(\beta) = \beta$.*

Proof. By a folklore result (see [23, Exercise 2.7]), every normal function on the ordinals has arbitrarily large fixed points. By inspection of the proof, it follows that this holds for normal functions $g\colon \lambda \to \lambda$ whenever $\mathrm{cf}(\lambda) > \omega$. Applying this to ℓ_x yields the desired result. □

The significance of this result is in the fact that it implies that for every Type 2 computation there is δ such that at time step δ, the machine has written δ bits on the output tape. In particular this means that the initial segment of the output of length δ depends only on (at most) $x{\restriction}\delta$. This is a feature which is not found in Type 2 computations of length ω and it can be useful in proving what T2κTM can and cannot do (see, e.g., Proposition 5.3.6).

We close the section with a result summarising well known results on the Type 2 computability of familiar functions:

Proposition 1.2.28. *The following functions are Type 2 computable:*

(a) the functions $\langle \ldots \rangle_n \colon (\kappa^\kappa)^n \to \kappa^\kappa$ defined in Section 1.2.2,

(b) the function $c\colon \kappa^\kappa \times \kappa \to \kappa^\kappa$ defined as $c(p, \alpha) = p_\alpha$,

(c) for any Type 1 computable function $f\colon \kappa \to \kappa$, the function $f'\colon \kappa^\kappa \to \kappa^\kappa$ defined as $f'(p)(\alpha) = f(p(\alpha))$,

(d) for any Type 2 computable function $g\colon \kappa^\kappa \to \kappa^\kappa$, the function $g'\colon \kappa^\kappa \to \kappa^\kappa$ defined as

$$g'(\langle (p_\alpha) \rangle_{\alpha\in\kappa}) = \langle (g(p_\alpha))_{\alpha\in\kappa} \rangle.$$

This proposition follows by a straightforward proof based on the computability of the pairing function $\ulcorner \cdot \urcorner$. In light of this result, when specifying algorithms we will often tacitly exploit the computability of these operations on sequences.

1.2.5 Limit machines

We close our section on computability by mentioning another computational model, which we will need to state the Mind Change Principle in Section 4.2.
Intuitively, a limit κ-Turing machine is a T2κTM which is allowed to revise its output during its computation for fewer than κ times, with the condition that each cell of the output tape eventually stabilizes to some value $s \in \{0,1\}$. Formally this can be modeled with our usual "append only" κ-Turing Machines by adding the symbols C to our standard $\{0,1\}$ alphabet and tweaking our interpretation of the output of such machines. A function $f :\subseteq 2^{<\kappa} \cup 2^{\kappa} \to 2^{<\kappa} \cup 2^{\kappa}$ is *computed by the limit κ-Turing Machine M* if for all $x \in 2^{<\kappa} \cup 2^{\kappa}$, M computes indefinitely on x if and only if $x \in \mathrm{dom}(f)$ and produces an output $M(x) \in \{0,1,C\}^{\kappa} \cup \{0,1,C\}^{<\kappa}$ such that there exists $\beta \in \kappa$ with $M(x)(\alpha) \neq C$ for all $\alpha \geq \beta$ and if γ is the least ordinal such that $M(x)(\alpha) \neq C$ for all $\alpha \geq \gamma$, the string $(M(x)(\alpha))_{\gamma \leq \alpha < \kappa}$ corresponds to $f(x)$. In this case, we define the *number of mind changes* in the computation for $M(x)$ as $\delta = \mathrm{ot}\{\alpha \in \gamma \mid M(x)(\alpha) = C\}$ and we denote it as $\mathrm{mc}(M,x)$.

Definition 1.2.29 (Mind Changes). *We say that $f :\subseteq 2^{<\kappa} \cup 2^{\kappa} \to 2^{<\kappa} \cup 2^{\kappa}$ is* computable with β mind changes *if and only if there exists a limit κ-Turing Machine M computing f with $\beta \geq \sup_{x \in \mathrm{dom}(f)}\{\mathrm{mc}(M,x)\}$.*

By the usual coding, we can transfer these notions of computability to functions of type $f :\subseteq \kappa^{<\kappa} \cup \kappa^{\kappa} \to \kappa^{<\kappa} \cup \kappa^{\kappa}$.

Lemma 1.2.30. *Let $\beta \in \kappa$ and let $f :\subseteq \kappa^{\kappa} \to \kappa^{\kappa}$ be computable with β mind changes. Let $g :\subseteq \kappa^{\kappa} \to \kappa^{\kappa}$ be a Type 2 κ-computable function. The compositions $g \circ f$ and $f \circ g$ are computable with β mind changes.*

Proof. Let M be the limit κ-Turing Machine computing f and let N be the ordinary κ-Turing Machine computing g. It is straightforward to see that the limit machine obtianed by composing M and N computes $f \circ g$ with β mind changes. For $g \circ f$, we consider the machine M' obtained from M and N again by composition, with the additional requirement that whenever M changes its mind, we restard the computation for N as well. As above, M' computes the function $g \circ f$ with β mind changes. □

Any κ-computable function is limit computable. We mention that, as in the classical case, the computational power of limit machines increases with the number of mind changes:

Example 1.2.31. *The function $f\colon \kappa^{\kappa} \to \kappa$ defined as*

$$f(0^{\kappa}) = 0,\ f(p) = 1\ \ \forall p \in \kappa^{\kappa} \setminus \{0\}$$

is computable with 1 mind change, but it is not computable.

The function f in the example above is sometimes called *equality test for* 0, and it is equivalent to the function LPO_{κ} which we will introduce in Chapter 3. In Proposition 4.2.2 we will see that for every $\beta \in \kappa$ there are functions which are computable with β mind changes but for every $\alpha < \beta$, they are not computable with α mind changes.

1.3 Computable analysis

We give an overview of the main results in generalized computable analysis underlying the theory of Weihrauch degrees, and we introduce some operations on degrees which will be useful for the classification results in later chapters. Generalized computable analysis was started by Galeotti in [16] and further developed Galeotti and Nobrega in [19]. The definitions and results presented in this section are entirely analogous to the classical ones, and they are mainly taken from [16]. For a complete introduction to computable analysis, see [41]. For the classical analogues of the operations on degrees which we introduce, see [6].

We define the concepts of notations and representations. Intuitively, these are the codings necessary to transport notions of computability on arbitrary spaces.

Definition 1.3.1 (Represented spaces). *Let X be any set. A* notation *for X is a partial surjective map $\nu :\subseteq \kappa \to X$, a* representation *for X is a partial surjective map $\delta :\subseteq \kappa^\kappa \to X$. A pair (X, δ) where δ is a representation for X is called a* represented space*.*

In the context of represented spaces, the letter ν will always be used for notations, and the letter δ will always be used for representations.
Given a represented space (X, δ_X), $p \in \mathrm{dom}(\delta_X)$ and $x \in X$, if $\delta_X(p) = x$ we say that p is *a name* or *a code* for x. We say that x is a *computable element* of the represented space (X, δ_X) if it has a computable name. Accordingly we say that x has degree $\leq_{\mathrm{T}} \boldsymbol{d}$ if it has a name with Turing degree $\boldsymbol{d}$.

We can form new represented spaces from old ones by employing some familiar techniques:

Definition 1.3.2 (Subspaces, products and sequence spaces). *Let $(X, \delta_X), (Y, \delta_Y)$ be represented spaces and let (Z, ν_Z) be a set equipped with a notation. We define:*

- *the* product space representation *of $X \times Y$ as $\delta_{X\times Y}(p) = (x, y)$ if and only if $p = \langle q, s\rangle$ with $\delta_X(q) = x$ and $\delta_Y(s) = y$,*
- *the* infinite product space representation *of X^β, for any $\beta \in \kappa$, as $\delta_{X^\beta}(p) = (x_\alpha)_{\alpha\in\kappa}$ if and only if $\delta_X(p_\alpha) = x_\alpha$ for all $\alpha \in \beta$,*
- *the* sequence space representation *of X^κ as $\delta_{X^\kappa}(p) = (x_\alpha)_{\alpha\in\kappa}$ if and only if $\delta_X(p_\alpha) = x_\alpha$ for all $\alpha \in \kappa$,*
- *the* representation for the set of (δ_X, δ_Y)-continuous functions $[X \to Y]$ *as $\delta_{[X\to Y]}(p) = f$ if and only if $p = 0^n1p'$ where $n \in \omega$ is the code for a Turing program P which, with oracle $p' \in 2^\kappa$, computes a continuous realizer $F :\subseteq \kappa^\kappa \to \kappa^\kappa$ of f,*[9]
- *the* sequence space representation *$\delta_{Z^\kappa}(p) = (x_\alpha)_{\alpha\in\kappa}$ if and only if $\nu_Z(p(\alpha)) = x_\alpha$ for all $\alpha \in \kappa$,*
- *if $A \subseteq X$, the* subspace representation *δ_A as δ_X corestricted to A, in symbols $\delta_X{\restriction}\delta_X^{-1}[A]$.*

When clear from context, we will drop mention of the representation associated to a given represented set. Similarly, when not explicitly stated, we always assume that product spaces, subspaces and sequence spaces are equipped with the representations defined above.

We will frequently need to work with *partial multifunctions* $f :\subseteq X \rightrightarrows Y$ (we will sometimes also use the name *operations*). Set-theoretically, a partial multifunction between X and Y is just a subset of $X \times Y$, so a relation. Nonetheless, seeing these objects as functions is more in line with the way we will use them. Formally, a partial multifunction $f :\subseteq X \rightrightarrows Y$ is determined by its domain $\mathrm{dom}(f) \subseteq X$ and by a function $F\colon \mathrm{dom}(f) \to \mathcal{P}(Y)$ associating to each $x \in \mathrm{dom}(f)$ the set of its images under f. Throughout this thesis, we will write $f(x)$ instead of $F(x)$. Obviously, total multifunctions, partial functions and total functions are special cases of partial multifunctions. Most of the theory is developed for partial multifunctions as this is the type of relations which is better suited to the needs of computable analysis.

Note that it is in principle not straightforward to define the *composition* of two multi-valued operations, as multi-valuedness as well as non-totality make the usual definition of functional composition not appropriate. Moreover, the definition of composition which is used in computable analysis is tailor-made so that it goes along well with the notions of realizers and Weihrauch reducibility. Let $f :\subseteq X \rightrightarrows Y$ and $g :\subseteq Y \rightrightarrows Z$ be partial multifunctions, we define $g \circ f$ as:

$$g(f(x)) = \bigcup_{y\in f(x)} g(y),$$

where $\mathrm{dom}(g \circ f) = \{x \in \mathrm{dom}(f) \mid f(x) \subseteq g\}$. Note that this does not coincide with the most commonly used notion of relational composition.

[9]This representation is surjective in light of Proposition 1.2.22.

For clarity we also define here the meaning of *preimage* in the context of partial multivalued functions: if $f :\subseteq X \rightrightarrows Y$ and $U \subseteq Y$, we define $f^{-1}[U]$ as

$$f^{-1}[U] = \{x \in \mathrm{dom}(f) \mid f(x) \cap U \neq \emptyset\}.$$

Definition 1.3.3 (Realizer)**.** *Let (X, δ_X) and (Y, δ_Y) be represented spaces and let $f :\subseteq X \rightrightarrows Y$ be a partial multifunction. A partial function $F :\subseteq \kappa^\kappa \to \kappa^\kappa$ is a* realizer *of f if and only if the following diagram commutes:*

$$\begin{tikzcd}
\kappa^\kappa \arrow[r, "F"] \arrow[d, "\delta_X"'] & \kappa^\kappa \arrow[d, "\delta_Y"] \\
X \arrow[r, "f"] & Y
\end{tikzcd}$$

in the sense that for every $p \in \mathrm{dom}(f \circ \delta_X)$ we have $\delta_Y(F(p)) \in f(\delta_X(p))$.

Note that we impose no other requirement on realizers, in particular, F need not be continuous nor computable. Moreover, F does not need to be uniform in the codes: if $p, p' \in \kappa^\kappa$ are distinct sequences such that $\delta_X(p) = \delta_X(p') = x$ and $y, y' \in f(x)$ are two distinct elements of Y, we allow the case where $\delta_Y(F(p)) = y$, $\delta_Y(F(p')) = y'$. In particular this means that the realizer F does not necessarily induce a choice function on the relation f.

Definition 1.3.4 (δ_X, δ_Y-computability and continuity)**.** *A partial multifunction $f :\subseteq (X, \delta_X) \rightrightarrows (Y, \delta_Y)$ is called (δ_X, δ_Y)-*computable *(risp.* continuous*) if it admits a Type 2 computable (risp. continuous) realizer.*

We can now introduce the notion of Weihrauch reducibility and Weihrauch completeness:

Definition 1.3.5 (Weihrauch reducibility)**.** *Let $f :\subseteq X \rightrightarrows Y$ and $g :\subseteq U \rightrightarrows V$ be operations on represented spaces, we say that f is* Weihrauch reducible *to g, in symbols $f \leq_{\mathrm{W}} g$ if there exist two Type 2 computable functions $H, K :\subseteq \kappa^\kappa \to \kappa^\kappa$ such that if G is any realizer of g, the function*

$$p \mapsto H(\langle p, G(K(p))\rangle)$$

is a realizer of f.

As variations of the same concept, we will say that f *topologically Weihrauch reduces to G* ($f \leq_{\mathrm{tW}} g$) if H, K are continuous; f *Weihrauch reduces to G relative to oracle $o \in \kappa^\kappa$* ($f \leq^o_{\mathrm{W}} g$) if H, K are computable with respect to the oracle o and f *strongly Weihrauch reduces to G* ($f \leq_{\mathrm{sW}} g$) if H, K are computable functions such that for every G realizing g, the function $H \circ G \circ K$ realizes f. By definition it is clear that for any f, g and any oracle o:

$$f \leq_{\mathrm{sW}} g \implies f \leq_{\mathrm{W}} g \implies f \leq^o_{\mathrm{W}} g \implies f \leq_{tW} g.$$

Definition 1.3.6 (Weihrauch completeness)**.** *Let M be any set of operations between represented spaces and let $f \in M$. We say that f* is Weihrauch complete for M with respect to $\leq_{\mathrm{W}}$ *if for every $g \in M$ we have $g \leq_{\mathrm{W}} f$. Similarly we say f is Weihrauch complete for M with respect to $\leq_{\mathrm{tW}}$ if for every $g \in M$ we have $g \leq_{\mathrm{tW}} f$.*

If $f \leq_{\mathrm{W}} g$ and $g \leq_{\mathrm{W}} f$, we write $f \equiv_{\mathrm{W}} g$ and we say that f is *Weihrauch equivalent* to g. It is straightforward to see that $\leq_{\mathrm{W}}$ is a preorder and hence $\equiv_{\mathrm{W}}$ is an equivalence relation. Equivalence classes of the $\equiv_{\mathrm{W}}$ relations are called *Weihrauch degrees*, and we will often denote the Weihrauch degree of an operation f as $[f]$. One of the objectives of computable analysis is characterising the Weihrauch degree of functions which correspond, via the coding of theorems we described in the introduction, to results mathematical analysis.

We introduce a notion of reduction between different representations of the same set.

Definition 1.3.7 (Reduction between representations)**.** *Let X be a set and let $\delta, \delta' :\subseteq \kappa^\kappa \to X$ be two representations of X, then we say that δ* continuously reduces to *δ', in symbols $\delta \leq_{\mathrm{t}} \delta'$, if id_X is (δ, δ')-continuous and δ* computably reduces to *δ', in symbols $\delta \leq_{\mathrm{c}} \delta'$ if id_X is (δ, δ')-computable. If $\delta \leq_{\mathrm{t}} \delta'$ and $\delta' \leq_{\mathrm{t}} \delta$ (risp. $\delta \leq_{\mathrm{c}} \delta'$ and $\delta' \leq_{\mathrm{c}} \delta$), we say that δ is* continuously equivalent to *(risp.* computably equivalent to*) δ' and we denote this as $\delta \equiv_{\mathrm{t}} \delta'$ (risp. $\delta \equiv_{\mathrm{c}} \delta'$).*

It is straightforward to see that if δ_X, δ'_X are two representations of X such that $\delta'_X \leq_{\mathrm{t}} \delta_X$ (risp. $\delta'_X \leq_{\mathrm{c}} \delta_X$) and δ_Y, δ'_Y are two representations of Y such that $\delta_Y \leq_{\mathrm{t}} \delta'_Y$ (risp. $\delta_Y \leq_{\mathrm{c}} \delta'_Y$) and $f :\subseteq X \to Y$ is (δ_X, δ_Y) continuous (risp. computable), then f is (δ'_X, δ'_Y) continuous (risp. computable).

Further we define the property of admissibility for a given representation:

Definition 1.3.8 (Admissible representation)**.** *Let (X, τ) be a topological space and let $\delta :\subseteq \kappa^\kappa \to X$ a representation. The representation δ is called* admissible *if it is continuous with respect to τ and for every continuous function $\varphi :\subseteq \kappa^\kappa \to X$, there exists a continuous function $h :\subseteq \kappa^\kappa \to \kappa^\kappa$ such that $\varphi(x) = \delta(h(x))$.*

The definition of admissibility in particular implies that if δ is admissible and δ' is any other continuous representation of X, $\delta' \leq_{\mathrm{t}} \delta$, in other words δ is maximal with respect to $\leq_{\mathrm{t}}$ among the continuous representations of X. We will often call a pair (X, δ) where X is a topological space and δ is an admissible representation an *admissibly represented* space.

We can now state the Main Theorem of Generalized Computable Analysis [16, Theorem 4.2.13]. It is the generalized computable analysis equivalent of the Main Theorem of Computable Analysis due to Weihrauch.

Theorem 1.3.9. *Let (X, τ), (Y, ξ) be topological spaces, let δ_X, δ_Y be admissible representations of X and Y respectively, and let $f :\subseteq X \to Y$ be a function, then*

$$f \text{ is } (\delta_X, \delta_Y)\text{-continuous} \iff f \text{ is continuous.}$$

The Main Theorem of Generalized Computable Analysis in particular implies that the space $[X \to Y]$ of Definition 1.3.2 coincides with the space of (topologically) continuous functions between two admissibly represented topological spaces X and Y, so the representation given there is an adequate tool to describe continuous function spaces.
We now turn to stating some basic results which we are going to employ for our Weihrauch classification results in later chapters.

Proposition 1.3.10. *Let $f :\subseteq X \rightrightarrows Y$ be an operation on represented spaces and assume $g :\subseteq Y \to Z$ is a single-valued operation between represented spaces with a computable realizer, then $g \circ f \leq_{\mathrm{sW}} f$.*

Proof. Let $G :\subseteq \kappa^\kappa \to \kappa^\kappa$ be a computable function realizing g, then for any realizer F of f, the function $G \circ F \circ \mathrm{id}_{\kappa^\kappa}$ is a realizer of $g \circ f$, showing the required strong reduction. □

Clearly, the same holds when composing on the right, so $f \circ g \leq_{\mathrm{sW}} f$.

Corollary 1.3.11. *In the assumptions of Proposition 1.3.10, if g is additionally an injection with computable left inverse, then $f \equiv_{\mathrm{sW}} g \circ f$. Moreover, in the same assumptions, if g surjects onto* $\mathrm{dom}(f)$ *and it has a computable right inverse, then $f \circ g \equiv_{\mathrm{sW}} f$.*

This justifies studying the Weihrauch degree of some operation f by considering $g \circ f$ or $f \circ g$ where g is an appropriate computable function.

We introduce some well known operations on degrees and state some of their properties. The proofs of the properties stated depend only on the computability of various pairing functions and projection functions. We give reference to the proofs for their classical counterparts, noting that the same exact proofs can be carried out in the generalized context. For the remainder of this section, assume that generalized Baire space is equipped with the representation $\mathrm{id}_{\kappa^\kappa}$.

Definition 1.3.12 (Product). *Let $f :\subseteq X \rightrightarrows Y$ and $g :\subseteq U \rightrightarrows V$ be multi-valued operations between represented spaces and define the* product $f \times g$ *as*

$$(f \times g)(x, u) = f(x) \times g(u),$$

with $\mathrm{dom}(f \times g) = \mathrm{dom}(f) \times \mathrm{dom}(g)$.

Proposition 1.3.13 (Monotonicity of products). *Let $f \leq_{\mathrm{W}} f'$, $g \leq_{\mathrm{W}} g'$ be functions between represented spaces, then $f \times f' \leq_{\mathrm{W}} g \times g'$ and the same holds with respect to strong Weihrauch reducibility*

Proof. See [6], Proposition 3.2. □

This in particular means that if $f \equiv_{\mathrm{W}} f'$ (risp. $f \equiv_{\mathrm{sW}} f'$) and $g \equiv_{\mathrm{W}} g'$ ($g \equiv_{\mathrm{sW}} g'$), then $f \times g \equiv_{\mathrm{W}} f' \times g'$ ($f \times g \equiv_{\mathrm{sW}} f' \times g'$), hence we can define a product operation on the (strong) Weihrauch degrees as $[f] \times [g] = [f \times g]$. It can also be shown that the product operation is associative and commutative up to strong Weihrauch equivalence, while the identity function on generalized Baire space is a neutral element for the product up to ordinary Weihrauch equivalence (see [6, Proposition 3.7]).

Definition 1.3.14 (Cylindrification). *Let $f :\subseteq X \rightrightarrows Y$ be a multi-valued operation between represented spaces, define the* cylindrification *of f as $\mathrm{id}_{\kappa^\kappa} \times f$. We call f a* cylinder *if it is equivalent to its cylindrification, i.e.,$f \equiv_{\mathrm{sW}} \mathrm{id}_{\kappa^\kappa} \times f$.*

Note that for any operation f, trivially $f \equiv_{\mathrm{W}} \mathrm{id}_{\kappa^\kappa} \times f$ and $f \leq_{\mathrm{sW}} \mathrm{id}_{\kappa^\kappa} \times f$, whereas $\mathrm{id}_{\kappa^\kappa} \times f \leq_{\mathrm{sW}} f$ (and hence $f \equiv_{\mathrm{sW}} \mathrm{id}_{\kappa^\kappa} \times f$) does not always hold (hence not all operations are cylinders).[10]

Proposition 1.3.15. *Let f and g be two multi-valued operations between represented spaces. Then*

$$f \leq_{\mathrm{W}} g \iff \mathrm{id}_{\kappa^\kappa} \times f \leq_{\mathrm{sW}} \mathrm{id}_{\kappa^\kappa} \times g.$$

Proof. See [6, Proposition 3.5]. □

This (together with $g \leq_{\mathrm{sW}} \mathrm{id}_{\kappa^\kappa} \times g$ for all g) immediately gives:

Corollary 1.3.16 (Reduction to cylinders). *Let f be a cylinder, g any multi-valued operation between represented spaces, then*

$$g \leq_{\mathrm{W}} f \iff g \leq_{\mathrm{sW}} f.$$

This last result will prove useful in settings where we know that some operation f is a cylinder and we want to prove $g \not\leq_{\mathrm{W}} f$ for some particular g. In such cases it is sufficient to rule out the possibility of a strong reduction between g and f to obtain the non-existence of any reduction. Similarly, if we want to show a strong reducibiltity to a cylinder, it suffices to come up with computable functions witnessing an ordinary Weihrauch reduction.

Lastly, we can define the operation of parallelization of a given operation f, intuitively this corresponds to taking κ-many copies of f.

Definition 1.3.17 (Parallelization). *Let $f :\subseteq X \rightrightarrows Y$ be a multi-valued operation, define its* parallelization $\prod f :\subseteq X^\kappa \rightrightarrows Y^\kappa$ *as*

$$\prod f((p_\alpha)_{\alpha\in\kappa}) = \{(q_\alpha)_{\alpha\in\kappa} \mid \forall\alpha \in \kappa(q_\alpha \in f(p_\alpha))\},$$

i.e., $\prod f((p_\alpha)_{\alpha\in\kappa}) = \prod_{\alpha\in\kappa} f(p_\alpha)$.

[10] Consider for example the function $f: \kappa^\kappa \to \kappa^\kappa$ with constant value 0^κ, it is trivial to see that $\mathrm{id}_{\kappa^\kappa} \times f \not\leq_{\mathrm{sW}} f$ as that would require the existence of two computable functions $H, K :\subseteq \kappa^\kappa \to \kappa^\kappa$ such that $H \circ f \circ K$ realizes $\mathrm{id}_{\kappa^\kappa} \times f$, but by definition of f, $H \circ f \circ K$ would be the function with domain $\mathrm{dom}(K)$ and constant value $H(0^\kappa)$.

When speaking about parallelized operations, we always assume that the spaces X^κ and Y^κ are endowed with the sequence space representations δ_X^κ and δ_Y^κ.

We remark that the notation introduced is not the standard notation for the parallelization of f, which would normally be denoted as $\widehat{f}$. Note that there is a potential risk of confusion of this notation with the usual notation Cartesian product of sets, which also uses the symbol $\prod$. We follow the convention that the symbol $\prod$, in the context of a Cartesian product of sets, will always have subscripts indicating the index set over which the product is taken. Occurrences of the $\prod$ symbol without indices will refer to parallelization.

Proposition 1.3.18 (Properties of parallelization)**.** *Let f, g be operations on represented spaces, we have*

(a) $f \leq_{\mathrm{sW}} \prod f$ (parallelization is extensive),

(b) $f \leq_{\mathrm{W}} g \implies \prod f \leq_{\mathrm{W}} \prod g$ (parallelization is monotone),

(c) $\prod f \equiv_{\mathrm{sW}} \prod(\prod f)$ (parallelization is idempotent).

Monotonicity holds also for strong reducibility.

Proof. See [6, Proposition 4.2]. $\square$

Lemma 1.3.19. *Let (A, δ_A), (B, δ_B) be represented spaces and define the functions $h_A \colon A^\kappa \times A^\kappa \to A^\kappa$ given by*

$$h((a_\alpha)_{\alpha\in\kappa}, (a'_\alpha)_{\alpha\in\kappa})(\beta) = a_\gamma \text{ if } \beta \text{ is the } \gamma\text{-th even ordinal,}$$

$$h((a_\alpha)_{\alpha\in\kappa}, (a'_\alpha)_{\alpha\in\kappa})(\beta) = a'_\gamma \text{ if } \beta \text{ is the } \gamma\text{-th odd ordinal,}$$

and $h_{A,B} \colon A^\kappa \times B^\kappa \to (A \times B)^\kappa$ given by

$$h(((a_\alpha)_{\alpha\in\kappa}), ((b_\alpha)_{\alpha\in\kappa})) = (a_\alpha, b_\alpha)_{\alpha\in\kappa}.$$

Both h_A and $h_{A,B}$ are computable bijections with computable inverses.

The proof is straightforward and essentially only requires the computability of the pairing functions. The following results are immediate consequences of Lemma 1.3.19.

Proposition 1.3.20. *Let $f :\subseteq X \rightrightarrows y$ and $g :\subseteq U \rightrightarrows V$ be operations on represented spaces, then*

$$\prod f \times g \equiv_{\mathrm{sW}} \prod f \times \prod g.$$

Proposition 1.3.21. *Let $f :\subseteq X \rightrightarrows Y$ be an operation, then*

$$\prod f \equiv_{\mathrm{sW}} \prod f \times \prod f.$$

Finally, we obtain a sufficient condition for being a cylinder:

Corollary 1.3.22. *Let f be an operation on represented spaces, and assume* $\mathrm{id} \leq_{\mathrm{sW}} \prod f$. *Then $\prod f$ is a cylinder.*

Proof. We have $\mathrm{id}_{\kappa^\kappa} \leq_{\mathrm{sW}} \prod f$ by assumption, hence $\mathrm{id}_{\kappa^\kappa} \times \prod f \leq_{\mathrm{sW}} \prod f \times \prod f$ by monotonicity of the product, and $\prod f \times \prod f \equiv_{\mathrm{sW}} \prod f$ by Proposition 1.3.21. This shows that $\mathrm{id}_{\kappa^\kappa} \times \prod f \leq_{\mathrm{sW}} \prod f$, hence $\prod f \equiv_{\mathrm{sW}} \mathrm{id}_{\kappa^\kappa} \times \prod f$, which is what was to show. $\square$

We mention here that all results on products, parallelizations and cylindrification mentioned in this section relativize to any oracle, i.e.,for every $o \in \kappa^\kappa$, we can replace $\leq_{\mathrm{W}}$ and $\leq_{\mathrm{sW}}$ with $\leq_{\mathrm{W}}^o$ and $\leq_{\mathrm{sW}}^o$ (and accordingly the term *computable function* with *o-computable function*) in any result between Proposition 1.3.13 and Corollary 1.3.22. This is seen at once by inspection of the proofs of these statements.

1.4 Surreal numbers and $\mathbb{R}_\kappa$

We introduce Conway's surreal numbers No, which are necessary to define the generalized real line $\mathbb{R}_\kappa$, introduced by Galeotti in [16]. As mentioned in the introduction, the field $\mathbb{R}_\kappa$ shares many features with $\mathbb{R}$, and developing analytical results for $\mathbb{R}_\kappa$ is an important part of *generalised analysis* (see [16], [17], [11], [18]). The generalized real line will be the setting of generalized computable analysis.

We mention here that, in the past, people have investigated other possible generalizations of the real line. In particular, Sikorski built a generalized field of rationals called $\mathbb{Q}^\kappa$ from the cardinal κ endowed with the Hessenberg operations in a way that is analogous to the construction of $\mathbb{Q}$ from $\mathbb{N}$. He then defined the field $\mathbb{R}^\kappa$ as the Cauchy completion of $\mathbb{Q}^\kappa$ (see [37]).[11] The field $\mathbb{R}^\kappa$ so obtained differs from $\mathbb{R}_\kappa$ as it has cardinality κ whereas $\mathbb{R}_\kappa$ has cardinality 2^κ. In [16], Galeotti explains how the cardinality of $\mathbb{R}^\kappa$ makes it unsuitable for computable analysis, and consequently he argues that $\mathbb{R}_\kappa$ is a more appropriate generalization of the real line. In line with this view, we will only consider $\mathbb{R}_\kappa$ in this thesis.

1.4.1 Definition and basic properties of surreal numbers

We define the class field of surreal numbers No introduced by Conway in [13] and state some of its properties. For more details on surreal numbers, we refer the reader to [21].

Definition 1.4.1 (Surreal numbers)**.** *Let* $\mathrm{No} = \{\alpha \to \{-,+\} \mid \alpha \in \mathrm{Ord}\}$*,* No *is called the class of* surreal numbers*.*

We refer to sequences $\alpha \to \{-,+\}$ as *sign sequences.* The *length* of a surreal number s, denoted $l(s)$ or $|s|$, is simply its domain. The advantage of the definition of surreals in terms of sign sequences is that we can define functions on the surreal by recursion on their length, in other words, if g is a function defined on sets of surreal numbers and φ is any property, definition of the type

$$f(x) = g(\{f(y) \mid l(y) < l(x) \wedge \varphi(y)\})$$

are justified by the Recursion Theorem on the ordinals. Another way of seeing surreal numbers is introducing a third sign $\uparrow$ which should be read as "undefined" and stipulating that a surreal number is a sequence $s\colon \mathrm{Ord} \to \{-,+,\uparrow\}$ which contains at least one $\uparrow$ symbol and in fact, if $\alpha = \min\{\delta \mid s(\delta) =\uparrow\}$, then $s(\beta) =\uparrow$ for all $\beta > \alpha$. We call such three-valued sequences *padded sign sequences.* Given a surreal number, it is straightforward to obtain the corresponding padded sign sequence, and viceversa. Padded sign sequences are useful for concisely defining the order on surreal numbers. First, we stipulate that our three signs are ordered as $- < \uparrow < +$.

Definition 1.4.2 (Order on No)**.** *Define* $\leq_s \subseteq \mathrm{No} \times \mathrm{No}$ *as follows: if* $x, y \in \mathrm{No}$*, let* s_x, s_y *be the corresponding padded sign sequences. We say* $x \leq_s y$ *if and only if* $x = y$ *or* $s_x(\alpha) < s_y(\alpha)$ *where* $\alpha = \min\{\beta \mid s_x(\beta) \neq s_y(\beta)\}$*.*

When clear from context, we will refer to $\leq_s$ as simply $\leq$.
The following theorem expresses the defining feature of the surreals, which is intuitively an extreme version of density.

Theorem 1.4.3 (Fundamental existence theorem)**.** *Let* L, R *be subset of* No *such that* $L < R$*, then there exists* $x \in \mathrm{No}$ *such that* $L < x < R$ *and for all* $y \in \mathrm{No}$*, if* $L < y < R$*, then* $x \subseteq y$*.*

Proof. See [21], Theorem 2.1. □

We denote the number x in the statement of the theorem as $[L|R]$. This is an example of another way to define surreals as *cuts.*[12] We say that a cut (L, R) is *cofinal* in a cut (L', R') if L is cofinal in L' and R is coinitial in R'. It is clear by definition that the relation of "being cofinal" is transitive. We have the following useful result:

[11] This construction was re-discovered recently by Asperó and Tsaprounis (see [3]).

[12] Whenever we refer to a pair of subsets of surreals as a cut, we assume that $L < R$ actually holds, similarly we only use the notation $[L|R]$ if (L, R) is a cut.

Lemma 1.4.4 (Cofinality lemma)**.** *Let (L, R), (L', R') be mutually cofinal cuts, then $[L|R] = [L'|R']$.*

Proof. See [21], Theorem 2.5 □

It turns out that every surreal number arises as the number $[L|R]$ for some cut (L, R), in particular each surreal number admits a canonical representation as a cut.

Definition 1.4.5 (Canonical cuts)**.** *Let $x \in \mathrm{No}$, the* canonical cut *of x, denoted (L_x, R_x) is given by*

$$L_x = \{y \subseteq x \mid y < x\}, \quad R_x = \{y \subseteq x \mid x > y\}$$

Lemma 1.4.6. *For all $x \in \mathrm{No}$, $x = [L_x \mid R_x]$*

Proof. See [21], Theorem 2.8. □

Exploiting the cut representation of the surreal numbers, it is possible to define surreal operations $+_s$ and $\cdot_s$ and, if we let $0 = \emptyset$ (the empty sequence) and $1 = (+)$, we obtain that the structure $(\mathrm{No}, +_s, \cdot_s, 0, 1, \leq_s)$ satisfies the axioms the theory of real closed fields. We use standard notation for the other common field operations such as additive and multiplicative inverses, and absolute values. As for the order symbol, we will often drop the subscript from the symbols for the field operations when the intended meaning is clear from context.

We now turn to define various subsets of the surreal numbers which will be useful in the construction of $\mathbb{R}_\kappa$.

Definition 1.4.7 (Notable subsets of No)**.** *For any ordinal α we define the sets:*

$$\mathrm{No}_\alpha = \{s \in \mathrm{No} \mid |s| = \alpha\}$$

$$\mathrm{No}_{<\alpha} = \{s \in \mathrm{No} \mid |s| < \alpha\}$$

$$\mathrm{No}_{\leq\alpha} = \mathrm{No}_\alpha \cup \mathrm{No}_{<\alpha}$$

We mention here that the real numbers are isomorphic to a subfield of the surreals contained in $\mathrm{No}_{\leq\omega}$, more precisely, if we let R be the set of surreals r of length $\leq\omega$ such that

$$|r| = \omega \implies r \text{ is not eventually constant}$$

then R a subfield of No and it is isomorphic to $\mathbb{R}$ ([21, Theorem 4.3]). From now on we identify R and $\mathbb{R}$, so we speak as real numbers as if they were elements of No. We also state the classical result (see [2], Corollary 1 on pg. 246):

Theorem 1.4.8. *Let λ be an uncountable regular cardinal, then $\mathrm{No}_{<\lambda}$ is a real closed field.*

We can identify the class of ordinals with a subclass of the surreals as follows: given an ordinal α, we identify it with the surreal number $(+)^\alpha$. It is known that the surreal operations $+_s, \cdot_s$ coincide with the Hessenberg (or natural) operations on ordinals (see [21], Theorems 4.5 and 4.6) and the surreal ordering coincides with the standard ordering on ordinals. We state this lemma for future reference, the proof is by an immediate application of the definition of canonical cuts.

Lemma 1.4.9. *Let $\alpha \in \mathrm{Ord}$, the canonical cut for α is given by $(L, \emptyset)$ where $L = \{(+)^\beta \mid \beta \in \alpha\}$.*

1.4.2 Conway Normal Form

We now present the derivation of the so-called Conway Normal Form for the surreal numbers. The material in this section is the basis for the results in Section 5.2.

Definition 1.4.10 (Archimedean classes)**.** *Let x and y be non-negative surreal numbers such that there exist $n, m \in \mathbb{N}$ with $n \cdot x > y$ and $m \cdot y > x$, we say that x and y lie in the same* Archimedean class *and we denote this as $x \sim_a y$. If u and v are any surreal numbers, we say $u \sim_a v$ if and only if $|u|_s \sim_a |v|_s$.*

It is obvious that $\sim_a$ is an equivalence relation on No. Moreover, we can define a relation $\ll$ as $x \ll y$ if and only if $x \not\sim_a y$ and $|x|_s < |y|_s$. The intuitive meaning of $x \ll y$ is then that, in magnitude, x is so much smaller than y that the set $\{z \cdot x \mid z \in \mathbb{Z}\}$ is contained in the surreal interval $(-y, y)$.

Definition 1.4.11 (ω-map)**.** *For any surreal number x we recursively define*

$$\omega^x = [0, r \cdot \omega^{x_L} \mid s \cdot \omega^{x_R}]$$

where s, r range over the positive real numbers, x_L ranges over L_x and x_R ranges over R_x.

Lemma 1.4.12. *Let x be any surreal number, then*

$$\omega^x = [0, r \cdot \omega^{x_L} \mid s \cdot \omega^{x_R}]$$

where r ranges over the set $\{n \mid n \in \omega\}$, s ranges over the set $\{1/n \mid n \in \omega\}$, x_L ranges over L_x and x_R ranges over R_x.

Proof. It is straightforward to see that the cut $(L', R') = (0, r \cdot \omega^{x_L} \mid s \cdot \omega^{x_R})$ where r ranges over the set $\{n \mid n \in \omega\}$ and s ranges over the set $\{1/n \mid n \in \omega\}$ and the cut (L, R) used in the definition of the ω-map are mututally cofinal, hence by Lemma 1.4.4, they define the same surreal number. $\square$

The ω-map behaves similarly to exponentiation, as can be seen in the following:

Lemma 1.4.13 (Properties of the ω-map)**.** *Let x and y be surreal numbers, then*

- $\omega^0 = 1$,
- $\omega^x > 0$,
- *if $x < y$, then $\omega^x \ll \omega^y$*
- $\omega^x \cdot \omega^y = \omega^{(x+y)}$.

Moreover, the ω-map coincides with exponentiation on ordinals, i.e.,the surreal number ω^α coincides with the ordinal number ω^α.

Proof. See [21], Theorem 5.2 and Theorem 5.4. $\square$

The ω-map also yields canonical representatives for Archimedean classes:

Lemma 1.4.14. *Let x be any surreal number, there exists a unique surreal number y of minimal length such that $x \sim_a y$ and $y = \omega^z$ for some $z \in$ No. Moreover, y is an initial segment of any element equivalent to it.*

Proof. See [21], Theorem 5.3. $\square$

We introduce transfinite sums of elements of the form $\omega^a \cdot r$ where r is a nonzero real number. These sums actually provide another characterization of surreal numbers which we will use in Chapter 5 for generalized computable analysis.

Definition 1.4.15 (Transfinite sums)**.** *Let γ be any ordinal, $(a_\alpha)_{\alpha\in\gamma}$ a sequence of strictly decreasing surreal numbers and $(r_\alpha)_{\alpha\in\gamma}$ a sequence of nonzero real numbers,*

- *if $\gamma = \beta + 1$ for some β, define*

$$\sum_{\alpha\in\gamma} \omega^{a_\alpha} \cdot r_\alpha = \left(\sum_{\alpha\in\beta} \omega^{a_\alpha} \cdot r_\alpha\right) + \omega^{a_\gamma} \cdot r_\gamma,$$

- *if γ is a limit, define*

$$\sum_{\alpha\in\gamma}\omega^{a_\alpha}\cdot r_\alpha = [L|R].$$

where

$$L = \Big\{\sum_{\alpha\in\beta+1}\omega^{a_\alpha}\cdot s_\alpha\Big\},$$

where β ranges in γ, $s_\alpha = r_\alpha$ for all $\alpha\in\beta$ and s_β ranges in $\mathbb{R}_{<r_\beta}$ and

$$R = \Big\{\sum_{\alpha\in\beta+1}\omega^{a_\alpha}\cdot s_\alpha\Big\},$$

where β ranges in γ, $s_\alpha = r_\alpha$ for all $\alpha\in\beta$ and s_β ranges in $\mathbb{R}_{>r_\beta}$.

Similarly to the proof of Lemma 1.4.12, by a mutual cofinality argument we can assume that in the limit cases, the coefficients s_β used to define the set L actually range in the set $\{s_\beta - 1/n \mid n\in\omega\}$ and the coefficients s_β used to define the set R range in the set $\{s_\beta + 1/n \mid n\in\omega\}$.

We define a *lexicographical ordering* on sequences of pairs of surreals and nonzero reals as follows: let $(a_\alpha, r_\alpha)_{\alpha\in\gamma}$ and $(a'_\alpha, r'_\alpha)_{\alpha\in\delta}$ be two such sequences and let $\beta = \min\{\xi \mid (a_\xi, r_\xi)\neq(a'_\xi, r'_\xi)\}$. If $\beta < \min\{\gamma,\delta\}$, then $(a_\alpha, r_\alpha)_{\alpha\in\gamma} > (a'_\alpha, r'_\alpha)_{\alpha\in\delta}$ if and only if $a_\beta \geq a'_\beta$ and $r_\beta \geq r'_\beta$; if $\beta=\gamma$, then $(a_\alpha, r_\alpha)_{\alpha\in\gamma} > (a'_\alpha, r'_\alpha)_{\alpha\in\delta}$ if and only if $r'_\beta < 0$ and lastly if $\beta=\delta$, then then $(a_\alpha, r_\alpha)_{\alpha\in\gamma} > (a'_\alpha, r'_\alpha)_{\alpha\in\delta}$ if and only if $r_\beta > 0$.

Theorem 1.4.16. *The expression $\sum_{\alpha\in\gamma}\omega^{a_\alpha}\cdot r_\alpha$ is defined for all strictly decreasing sequences of surreals $(a_\alpha)_{\alpha\in\gamma}$ and all sequences of nonzero real numbers $(r_\alpha)_{\alpha\in\gamma}$. Moreover,*

$$\sum_{\alpha\in\gamma}\omega^{a_\alpha}\cdot r_\alpha < \sum_{\alpha\in\delta}\omega^{a'_\alpha}\cdot r'_\alpha$$

if and only if $(a_\alpha, r_\alpha)_{\alpha\in\gamma} < (a'_\alpha, r'_\alpha)_{\alpha\in\delta}$ and lastly for any $(a_\alpha, r_\alpha)_{\alpha\in\gamma}$, any $\beta\in\gamma$ and any $\xi\in\beta$,

$$\left|\sum_{\alpha\in\gamma}\omega^{a_\alpha}\cdot r_\alpha - \sum_{\alpha\in\beta}\omega^{a_\alpha}\cdot r_\alpha\right| \ll \omega^{a_\xi}. \tag{1.1}$$

Proof. See [21], Theorem 5.5 □

In the following we refer to the property expressed in the inequality (1.1) as the *tail property.* Intuitively, it means that a sum is relatively close to every sum determined by an initial segment of the sequence its exponent-coefficient pairs, where relatively close is to be interpreted in terms of Archimedean classes.

Lemma 1.4.17. *Let $(a_\alpha)_{\alpha\in\gamma}$ be a strictly descending sequence of surreal numbers, $(r_\alpha)_{\alpha\in\gamma}$ a sequence of nonzero real numbers and let $x = \sum_{\alpha\in\gamma}\omega^{a_\alpha}\cdot r_\alpha$, then $l(x)\geq\gamma$.*

Proof. See [21], Theorem 5.6 □

Note that, while Gonshor describes such inequality as “crude”, we will show in Corollary 1.4.28 that the bound is actually optimal.

Theorem 1.4.18 (Conway Normal Form)**.** *For every surreal number x there exist unique sequences $(a_\alpha)_{\alpha\in\gamma}$ and $(r_\alpha)_{\alpha\in\gamma}$ such that*

$$x = \sum_{\alpha\in\gamma}\omega^{a_\alpha}\cdot r_\alpha.$$

Proof. See [21, Theorem 5.6] □

We remark that the Conway Normal Form of a number x is obtained from an approximation construction: one recursively build a sequence of exponent-coefficient pairs whose corresponding transfinite sums approximate the surreal number x with increasing accuracy. Later we will see that for particular surreal numbers, a similar procedure can be actually carried out on a T2κTM.

We state and prove a continuity property of Conway Normal Form which we will exploit in Section 5.2:

Lemma 1.4.19. *Let x, y and a be surreal numbers such that $0 \neq |x-y| \ll \omega^a$ and*

$$x = \sum_{\alpha\in\lambda} \omega^{e_\alpha} \cdot r_\alpha,\ y = \sum_{\alpha\in\mu} \omega^{f_\alpha} \cdot s_\alpha.$$

Let $\xi = \min\{\beta \mid (e_\beta, r_\beta) \neq (f_\beta, s_\beta)\}$, then $a > \max\{e_\xi, f_\xi\}$.

Proof. First, notice that it can't be the case that $\lambda + 1 = \mu + 1 = \xi$, because this would entail that the Conway Normal Forms of x and y coincide and $x = y$. Therefore we can assume at least one of e_ξ and f_ξ is defined. Assume only one of them is defined, say e_ξ, then by the tail property

$$\omega^{e_\xi} \gg \left| x - \sum_{\alpha\in\xi+1} \omega^{e_\alpha} \cdot r_\alpha \right| = \left| x - \left(\sum_{\alpha\in\xi} \omega^{e_\alpha} \cdot r_\alpha + \omega^{e_\xi} \cdot r_\xi \right) \right| = \left| x - (y + \omega^{e_\xi} \cdot r_\xi) \right|.$$

Therefore $|x-y| \sim_a \omega^{e_\xi} \ll \omega^a$, which implies that $a > e_\xi = \max\{e_\xi, f_\xi\}$. Lastly if both e_ξ and f_ξ are defined, consider z defined as

$$z = \sum_{\alpha\in\xi} \omega^{f_\alpha} \cdot s_\alpha,$$

and define $d_1 = x - z$ and $d_2 = y - z$. Using the tail property as above, we obtain $|d_2| \sim_a \omega^{f_\xi}$ and $|d_1| \sim_a \omega^{e_\xi}$. Now consider $d_1 - d_2$, there are two options: if $f_\xi \neq e_\xi$, then assume without loss of generality that $f_\xi > e_\xi$. It follows that $\omega^{f_\xi} \gg \omega^{e_\xi}$, hence $|d_1 - d_2| \sim_a \omega^{f_\xi}$. By assumption $|d_1 - d_2| = |x - y| \ll \omega^a$, so this immediately implies that $a > f_\xi = \max\{e_\xi, f_\xi\}$. If $f_\xi = e_\xi$, then it follows that $r_\xi \neq s_\xi$ and by an application of [21, Lemma 5.5] we obtain $d_1 = \omega^{e_\xi} \cdot r_\xi + t_1$, $d_2 = \omega^{e_\xi} \cdot s_\xi + t_2$ with $t_1 \ll \omega^{e_\xi}$ and $t_2 \ll \omega^{e_\xi}$. Hence $x - y = d_1 - d_2 = \omega^{e_\xi}(r_\xi - s_\xi) + t_1 - t_2 \sim \omega^{e_\xi}$, which again implies $e_\xi = \max\{e_\xi, f_\xi\} < a$. $\square$

We mention here that the definition of transfinite sum can be extended to include null real coefficients.

Definition 1.4.20 (Reduced sequences)**.** *Let $(e_\alpha, r_\alpha)_{\alpha\in\lambda}$ be a sequence of pairs of surreals with $(e_\alpha)_{\alpha\in\lambda}$ a strictly decreasing sequence of surreals and $(r_\alpha)_{\alpha\in\lambda} \subseteq \mathbb{R}$.*
We define the corresponding reduced sequence $(e_{\alpha_\beta}, r_{\alpha_\beta})_{\beta\in\gamma}$ *as the subsequence containing all (and only) the pairs with $r_\alpha \neq 0$.*

Lemma 1.4.21. *For all $(e_\alpha, r_\alpha)_{\alpha\in\lambda}$ with $(e_\alpha)_{\alpha\in\lambda}$ a strictly decreasing sequence of surreals and $(r_\alpha)_{\alpha\in\lambda} \subseteq \mathbb{R}$, we can meaningfully define*

$$\sum_{\alpha\in\lambda} \omega^{e_\alpha} \cdot r_\alpha$$

in the same way as we defined transfinite sums with sequences of nonzero coefficients. Moreover, if $(e_{\alpha_\beta}, r_{\alpha_\beta})_{\beta\in\gamma}$ is the reduced sequence associated to $(e_\alpha, r_\alpha)_{\alpha\in\lambda}$, we obtain that

$$\sum_{\alpha\in\lambda} \omega^{e_\alpha} \cdot r_\alpha = \sum_{\beta\in\gamma} \omega^{e_{\alpha_\beta}} \cdot r_{\alpha_\beta}$$

.

Proof. See [21, Lemma 5.4] $\square$

1.4.3 The generalized reals $\mathbb{R}_\kappa$

We introduce the field of κ-reals $\mathbb{R}_\kappa$, where most of the generalized computable analysis of this thesis will take place. The field $\mathbb{R}_\kappa$, as well as the field $\mathbb{Q}_\kappa$ of κ-rationals were first introduced in [16], where most of their basic properties are proven. We recall the definitions and properties relevant for the rest of the thesis, for more details we refer the reader to [16] and [17].

Definition 1.4.22 (The κ-rationals). *We define the* field of κ-rationals *as* $\mathbb{Q}_\kappa = \mathrm{No}_{<\kappa}$

By Theorem 1.4.8, $\mathbb{Q}_\kappa$ is a real closed field when equipped with the surreal operations and ordering, and by our assumption of $\kappa^{<\kappa} = \kappa$ we obtain $|\mathbb{Q}_\kappa| = \kappa$. Moreover, $\mathbb{Q}_\kappa$ is an η_κ-set, i.e.,whenever $L, R \subseteq \mathbb{Q}_\kappa$, $L < R$ and $|L \cup R| < \kappa$, there exists some $x \in \mathbb{Q}_\kappa$ such that $L < \{x\} < R$. Note that, if we identify the set $\{-, +\}$ with the set $2 = \{0, 1\}$, we immediately see that $\mathbb{Q}_\kappa$ is essentially the same as $2^{<\kappa}$, therefore in our assumptions there is a κ-computable function enumerating the κ-rationals.

Theorem 1.4.23. *The set $\mathbb{Q}_\kappa$ coincides with the set of surreals x with Conway Normal Form*

$$x = \sum_{\alpha \in \lambda} \omega^{e_\alpha} \cdot r_\alpha,$$

such that $\lambda < \kappa$, and for all $\alpha \in \lambda$, $e_\alpha \in \mathbb{Q}_\kappa$.

Proof. See [2], pages 242-245. □

Given any cut (L, R) with $L \cup R \subseteq \mathbb{Q}_\kappa$, we say that (L, R) is a *Cauchy cut* if L has no maximum and for all $\epsilon \in \mathbb{Q}_\kappa^+$ there exists $l \in L$, $r \in R$ such that $l + \epsilon > r$.

Definition 1.4.24 (The κ-reals). *We define the set $\mathbb{R}_\kappa = \mathbb{Q}_\kappa \cup \{[L|R] | (L, R) \text{is a Cauchy cut}\}$.*

Note that if $x \in \mathbb{R}_\kappa \setminus \mathbb{Q}_\kappa$, then $x = [L|R]$ for some Cauchy cut in $\mathbb{Q}_\kappa$, so in particular if $(s_\alpha)_{\alpha\in\kappa}$ is a sequence of κ-rationals which is entirely contained in L and is cofinal in it, then $x = \sup_{\alpha\in\kappa} s_\alpha$. It follows from the usual argument that one can always assume that such a sequence is strictly increasing.

Theorem 1.4.25. *The set $\mathbb{R}_\kappa$ endowed with the surreal operations and ordering is a real closed field extension of $\mathbb{R}$. Moreover, $\mathbb{R}_\kappa$ is Cauchy complete, an η_κ-set, κ-shperically complete, it has cardinality 2^κ, coinitiality and cofinality κ, and $\mathbb{Q}_\kappa$ is dense in $\mathbb{R}_\kappa$.*

Proof. See [16] Lemma 3.4.6, Lemma 3.4.7, Theorem 3.4.12, Theorem 3.4.14 for and Corollary 3.5.26. □

It can be shown that $\mathbb{R}_\kappa$ is the unique such field (see [19, Theorem 4]). It follows from the uniqueness theorem for $\mathbb{R}$ that $\mathbb{R}_\kappa$ is not Dedekind complete.

Theorem 1.4.26. *The Dedekind completion of $\mathbb{Q}_\kappa$ is included in $\mathrm{No}_{\leq\kappa}$. This implies in particular that $\mathbb{R}_\kappa \subseteq \mathrm{No}_{\leq\kappa}$.*

Proof. See [16, Lemma 3.4.13]. □

We prove a novel characterization of $\mathbb{R}_\kappa$ in terms of Conway Normal Form:

Theorem 1.4.27. *The set $\mathbb{R}_\kappa$ is given by $\mathbb{Q}_\kappa$ together with those surreals x of length κ which have Conway Normal Form*

$$x = \sum_{\alpha \in \kappa} \omega^{e_\alpha} \cdot r_\alpha,$$

such that $(e_\alpha)_{\alpha\in\kappa}$ is a coinitial sequence of elements of $\mathbb{Q}_\kappa$.

Proof. Let $x \in \mathbb{R}_\kappa \setminus \mathbb{Q}_\kappa$ and let $(e_\alpha, r_\alpha)_{\alpha\in\lambda}$ be its Conway Normal Form, i.e.,

$$x = \sum_{\alpha\in\lambda} \omega^{e_\alpha} \cdot r_\alpha.$$

By Lemma 1.4.17 we have $\lambda \leq \kappa = l(x)$. Now by density of $\mathbb{Q}_\kappa$ in $\mathbb{R}_\kappa$, there exists a sequence of κ-rationals $(q_\alpha)_{\alpha\in\kappa} \subseteq \mathbb{Q}_\kappa$ such that for all $\alpha \in \kappa$:

$$|x - q_\alpha| < \omega^{-\alpha-1} \ll \omega^{-\alpha}.$$

By Lemma 1.4.19, we obtain that the Conway Normal Form of x coincides with that of q_α up to exponents under $-\alpha$. By Theorem 1.4.23, the exponents in the Conway Normal Form of each q_α are κ-rationals, hence we obtain that $(e_\alpha)_{\alpha\in\kappa} \subseteq \mathbb{Q}_\kappa$. Therefore, since $x \notin \mathbb{Q}_\kappa$, again Theorem 1.4.23 implies that $\kappa \leq \lambda$, so we conclude that $\lambda = \kappa$. Now assume that $(e_\alpha)_{\alpha\in\kappa}$ is not coinitial in $\mathbb{Q}_\kappa$, so let $-\beta$ be a lower bound for it. By definition of the sequence $(q_\alpha)_{\alpha\in\kappa}$ we know that $|x - q_{\beta+1}| \ll \omega^{-\beta-1}$, so by Lemma 1.4.19 we obtain that the Conway Normal Forms of $q_{\beta+1}$ and x coincide up to exponents under $-\beta$. Since we assumed $-\beta$ to be a lower bound for $(e_\alpha)_{\alpha\in\kappa}$, this entails that the Conway Normal Form for x is an initial segment of the Conway Normal Form for q_β, which implies that q_β has a Conway Normal Form of length $\geq \kappa$. This is in contradiction with Theorem 1.4.23.

Conversely assume that $x \in \mathrm{No}$ has Conway Normal Form given by

$$x = \sum_{\alpha\in\kappa} \omega^{e_\alpha} \cdot r_\alpha,$$

where $(e_\alpha)_{\alpha\in\kappa}$ is a sequence of κ-rationals coinitial in $\mathbb{Q}_\kappa$. For all $\beta \in \kappa$, define

$$s_\beta = \sum_{\alpha\in\beta} \omega^{e_\alpha} \cdot r_\alpha.$$

Notice that the fact that $(e_\alpha)_{\alpha\in\kappa}$ is coinitial in $\mathbb{Q}_\kappa$ immediately implies that $(\omega^{e_\alpha})_{\alpha\in\kappa}$ is coinitial in $\mathbb{R}^+_\kappa$. By the tail property we obtain that for all $\beta \in \kappa$ and all $\eta, \xi > \beta$,

$$|s_\xi - s_\eta| < \omega^{e_\beta},$$

which, by the coinitiality of $(\omega^{e_\alpha})_{\alpha\in\kappa}$ in $\mathbb{R}^+_\kappa$, implies that $(s_\beta)_{\beta\in\kappa}$ is a Cauchy sequence of rationals, hence it converges to some κ-real s. Now an application of Lemma 1.4.19 implies that s must have the same Conway Normal Form of x, hence $x = s \in \mathbb{R}_\kappa$. ☐

As anticipated, this tangentially shows that the bound on the length of transfinite sums in Lemma 1.4.17 is optimal.

Corollary 1.4.28. *For all cardinals λ such that $\lambda^{<\lambda} = \lambda$, there are surreal numbers with Conway Normal Form*

$$x = \sum_{\alpha\in\lambda} \omega^{a_\alpha} \cdot r_\alpha$$

and $l(x) = \lambda$.

Proof. For any such λ, any λ-real which is not a λ-rational is an example of such a surreal number. ☐

In the rest of this thesis we will consider $\mathbb{R}_\kappa$ as a topological space, with topology generated by the open intervals with endpoints in $\mathbb{Q}_\kappa \cup \{+\infty, -\infty\}$. Note that intervals in $\mathbb{R}_\kappa$ need not have endpoints, as this property is equivalent to Dedekind completeness (see [34], Lemma 2.23). The same is true for closed and for open intervals. In the rest of this thesis, we will refer to intervals in $\mathbb{R}_\kappa$ with endpoints as *proper intervals* and to those without endpoints as *improper intervals.*[13]

We now turn to some preliminary definitions for computable analysis on $\mathbb{R}_\kappa$, starting with the introduction of a notation and two representations of $\mathbb{Q}_\kappa$ and the statement of some of their properties.

[13]Note that in the literature the terms proper and improper are sometimes used in a different way, i.e., to distinguish between intervals consiting of more than one point and intervals consisting of a single point.

Definition 1.4.29 (Notation for $\mathbb{Q}_\kappa$)**.** *Let $\nu_\kappa \colon \kappa \to \mathbb{Q}_\kappa$ be given by $\nu_{\mathbb{Q}_\kappa}(\alpha) = q$ if and only if $\nu(\alpha) = q'$, where ν is the computable enumeration of Corollary 1.2.13 and q is the sign sequence of length $|q'|$ with $q'(\beta) = 1 \iff q(\beta) = +$. When there is no risk of confusion, we will write $\overline{\alpha}$ for $\nu_{\mathbb{Q}_\kappa}(\alpha)$.*

Definition 1.4.30 (Representation of $\mathbb{Q}_\kappa$)**.** *Let $\delta_{\mathbb{Q}_\kappa} :\subseteq 2^\kappa \to \mathbb{Q}_\kappa$ be the representation given by $\delta_{\mathbb{Q}_\kappa}(p) = q$ if and only if $p = [\![w_\alpha]\!]$ where $w_\alpha = 00$ if $\alpha \in \operatorname{dom}(q)$ and $q(\alpha) = -$, $w_\alpha = 11$ if $\alpha \in \operatorname{dom}(q)$ and $q(\alpha) =$ and $w_\alpha = 10$ if $\alpha \notin \operatorname{dom}(q)$.*

Notice that $\delta_{\mathbb{Q}_\kappa}$ is injective, therefore every κ-rational has a unique $\delta_{\mathbb{Q}_\kappa}$-name. In light of this, we will sometimes say that a sequence $p \in \operatorname{dom}(\delta_{\mathbb{Q}_\kappa})$ is *the* code (or *the* name) for the rational $r = \delta_{\mathbb{Q}_\kappa}(p)$.

It is clear that, assuming Hypothesis 1.2.19, $\nu_{\mathbb{Q}_\kappa}$ is computationally equivalent to $\delta_{\mathbb{Q}_\kappa}$ in the following sense: there is a T2κTM which, on input $\alpha \in \kappa$, outputs $p' \in \operatorname{dom}(\delta_{\mathbb{Q}_\kappa})$ such that $\delta_{\mathbb{Q}_\kappa}(p') = \nu_{\mathbb{Q}_\kappa}(\alpha)$; conversely there is a κ-Turing Machine which, starting with any $p' \in \operatorname{dom}(\delta_{\mathbb{Q}_\kappa})$, terminates and outputs $\alpha \in \kappa$ such that $\delta_{\mathbb{Q}_\kappa}(p') = \nu_{\mathbb{Q}_\kappa}(\alpha)$.

We also introduce a representation of $\mathbb{Q}_\kappa$ based on the cut representation of surreal numbers:

Definition 1.4.31 (Cut representation of $\mathbb{Q}_\kappa$)**.** *Recursively define the following functions: $\delta^{\mathrm{c},0}_{\mathbb{Q}_\kappa}(p) = 0$ if and only if $p_\alpha = [\![10]\!]_{\beta<\kappa}$ for every $\alpha < \kappa$ and for evey $\alpha < \kappa$, define the function $\delta^{\mathrm{c},\alpha}_{\mathbb{Q}_\kappa}$ as $\delta^{\mathrm{c},\alpha}_{\mathbb{Q}_\kappa}(p) = [L|R]$ where*

- *for every $\alpha < \kappa$, $p_\alpha \in \bigcup_{\gamma<\alpha} \operatorname{dom}(\delta^{\mathrm{c},\gamma}_{\mathbb{Q}_\kappa}) \cup \{[\![10]\!]_{\beta\in\kappa}\}$*
- *for all even $\alpha < \kappa$, if $p_\alpha = [\![10]\!]_{\beta\in\kappa}$, then for every even $\beta > \alpha$, $p_\beta = [\![10]\!]_{\beta\in\kappa}$,*
- *for all odd $\alpha < \kappa$, if $p_\alpha = [\![10]\!]_{\beta\in\kappa}$, then for every odd $\beta > \alpha$, $p_\beta = [\![10]\!]_{\beta\in\kappa}$,*
- *$L = \{\delta^{\mathrm{c},\gamma}_{\mathbb{Q}_\kappa}(p_\beta) \mid \beta < \kappa$ is odd and $p_\beta \in \operatorname{dom}(\delta^{\mathrm{c},\gamma}_{\mathbb{Q}_\kappa})$ for some $\gamma < \alpha\}$,*
- *$R = \{\delta^{\mathrm{c},\gamma}_{\mathbb{Q}_\kappa}(p_\beta) \mid \beta < \kappa$ is odd and $p_\beta \in \operatorname{dom}(\delta^{\mathrm{c},\gamma}_{\mathbb{Q}_\kappa})$ for some $\gamma < \alpha\}$.*

Finally define $\delta^{\mathrm{c}}_{\mathbb{Q}_\kappa} = \bigcup_{\gamma\in\kappa} \delta^{\mathrm{c},\gamma}_{\mathbb{Q}_\kappa}$.

Notice that $\delta^{\mathrm{c}}_{\mathbb{Q}_\kappa}$ is a representation because for every $q \in \mathbb{Q}_\kappa$, there exists $p \in 2^\kappa$ such that p codes the canonical cut $[L_q|R_q]$.

Proposition 1.4.32. $\delta_{\mathbb{Q}_\kappa} \equiv_{\mathrm{c}} \delta^{\mathrm{c}}_{\mathbb{Q}_\kappa}$.

Proof. See [17], Lemma 4.14. □

We remark that the proof cited shows even more, namely that the transformation from $\delta_{\mathbb{Q}_\kappa}$ codes into $\delta^{\mathrm{c}}_{\mathbb{Q}_\kappa}$ codes always leads us to codes representing κ-rational numbers as canonical cuts. Since we can always transform any $\delta^{\mathrm{c}}_{\mathbb{Q}_\kappa}$-code into a $\delta_{\mathbb{Q}_\kappa}$ code, this entails that whenever we work with $\delta^{\mathrm{c}}_{\mathbb{Q}_\kappa}$ codes, we can assume that these represent a rational q by its canonical cut.

This result sheds light on the fact that both the representations of $\mathbb{Q}_\kappa$ introduced are really Type 1 objects. This is obvious for $\delta_{\mathbb{Q}_\kappa}$, as the entire sign sequence for the rational $\delta_{\mathbb{Q}_\kappa}(p)$ is contained in the initial segment of p up to the first 1010 bitstring, whereas the definition of $\delta^{\mathrm{c}}_{\mathbb{Q}_\kappa}$ is at first a bit too obscure to see this. Nonetheless, by Proposition 1.4.32, there is a κ-Turing Machine M which translates $\delta^{\mathrm{c}}_{\mathbb{Q}_\kappa}$-names into $\delta_{\mathbb{Q}_\kappa}$-names, hence running M on code p until M prints the first 1010 bitstring yields a Type 1 procedure which allows us to identify the κ-rational $\delta^{\mathrm{c}}_{\mathbb{Q}_\kappa}(p)$ in fewer than κ steps.

Proposition 1.4.33. *The field operations on $\mathbb{Q}_\kappa$ and the characteristic function of the order $<_{\mathbb{Q}_\kappa}$ are $\delta^{\mathrm{c}}_{\mathbb{Q}_\kappa}$-computable.*

Proof. See [17], Lemma 4.15. □

We are now ready to introduce three representations of $\mathbb{R}_\kappa$. Two of these were present in [17], whereas $\delta_{\mathbb{R}_\kappa}$ is a new representation which is more in line with the theory of $\mathbb{R}_\kappa$-metric spaces introduced in Chapter 2. It is clear from the definition that the latter is essentially the same as the two previous presentations, and we sketch a proof of their equivalence.

Definition 1.4.34 (Cauchy representations of $\mathbb{R}_\kappa$). *Define $\delta'_{\mathbb{R}_\kappa} :\subseteq 2^\kappa \to \mathbb{R}_\kappa$ as $\delta'_{\mathbb{R}_\kappa}(p) = r$ if and only if $\delta_{\mathbb{Q}_\kappa}(p_\alpha) = q_\alpha$ and for every $\alpha \in \kappa$, $x \in (q_\alpha - \frac{1}{\alpha+1}, q_\alpha + \frac{1}{\alpha+1})$ and define $\delta_{\mathbb{R}_\kappa} :\subseteq \kappa^\kappa \to \mathbb{R}_\kappa$ as $\delta_{\mathbb{R}_\kappa}(p) = r$ if and only if, letting $q_\alpha = \nu_{\mathbb{Q}_\kappa}(p(\alpha))$, $(q_\alpha)_{\alpha\in\kappa}$ converges to r and for all $\beta < \gamma < \kappa$, $|q_\beta - q_\gamma| < \frac{1}{\beta+1}$.*

The Cauchy representation $\delta_{\mathbb{R}_\kappa}$ is the representation which we obtain when considering $\mathbb{R}_\kappa$ as a computable κ-metric space (see Chapter 2).

Definition 1.4.35 (Veronese representation of $\mathbb{R}_\kappa$). *Define $\delta^{\mathrm{V}}_{\mathbb{R}_\kappa} :\subseteq 2^\kappa \to \mathbb{R}_\kappa$ as $\delta^{\mathrm{V}}_{\mathbb{R}_\kappa}(p) = x$ if and only if for each $\alpha \in \kappa$, $p_\alpha \in \mathrm{dom}(\delta_{\mathbb{Q}_\kappa})$ and $x = [L|R]$ where $L = \{\delta_{\mathbb{Q}_\kappa}(p_\alpha) \mid \alpha < \kappa \text{ even}\}$ and $R = \{\delta_{\mathbb{Q}_\kappa}(p_\alpha) \mid \alpha < \kappa \text{ odd}\}$ and for every $\alpha < \kappa$ we have $\delta_{\mathbb{Q}_\kappa}(p_{\alpha+1}) < \delta_{\mathbb{Q}_\kappa}(p_\alpha) + \frac{1}{\alpha+1}$.*

Proposition 1.4.36. *Under Hypothesis 1.2.19, the three representations $\delta^{\mathrm{V}}_{\mathbb{R}_\kappa}$, $\delta'_{\mathbb{R}_\kappa}$ and $\delta_{\mathbb{R}_\kappa}$ are all computably equivalent.*

Proof. See [17], Theorem 4.19 for a proof (which is not conditional on constructibility assumptions) that $\delta'_{\mathbb{R}_\kappa}$ is equivalent to $\delta^{\mathrm{V}}_{\mathbb{R}_\kappa}$. To prove that $\delta_{\mathbb{R}_\kappa}$ is in turn equivalent to both of them, it is then sufficient to notice that, modulo the computable transformations between $\nu_{\mathbb{Q}_\kappa}$ and $\delta_{\mathbb{Q}_\kappa}$ codes, if a sequence p is a valid $\delta^{\mathrm{V}}_{\mathbb{R}_\kappa}$-code for x, then it is a valid $\delta_{\mathbb{R}_\kappa}$-code for x and in turn if a sequence q is a valid $\delta_{\mathbb{R}_\kappa}$-code for y, then q is a $\delta'_{\mathbb{R}_\kappa}$-code for x. □

Proposition 1.4.37. *The representations $\delta_{\mathbb{R}_\kappa}$, $\delta^{\mathrm{V}}_{\mathbb{R}_\kappa}$ and $\delta'_{\mathbb{R}_\kappa}$ are all admissible with respect to the interval topology on $\mathbb{R}_\kappa$.*

Proof. Follows from the fact that $\delta_{\mathbb{R}_\kappa}$ is admissible as it is a Cauchy representation of a computable κ-metric space (cf. Chapter 2) and $\delta_{\mathbb{R}_\kappa} \equiv_{\mathrm{c}} \delta^{\mathrm{V}}_{\mathbb{R}_\kappa} \equiv_{\mathrm{c}} \delta'_{\mathbb{R}_\kappa}$ □

Proposition 1.4.38. *The field operations are $\delta'_{\mathbb{R}_\kappa}$-computable*

Proof. See [17], Theorem 4.17. □

Chapter 2

Computability on $\mathbb{R}_\kappa$-metric spaces

In the classical context, the notion of computable metric space dates back at least to Weihrauch (see [41]). The setup we introduce in this chapter is an adaptation of the classical framework. In particular we draw from [5], where the author introduces representations for spaces of Borel sets of computable metric spaces to obtain the notion of $\mathbf{\Sigma}^0_n$-effective measurability for functions between such spaces. Moreover, in [5], the author introduces a family of functions $(C_n)_{n\in\omega}$ and proves that each C_n is complete with respect to $\leq_{\mathrm{W}}$ in the class of $\mathbf{\Sigma}^0_n$-effectively measurable functions on Baire space. The goal of this chapter is to introduce the notion of computable $\mathbb{R}_\kappa$-metric spaces and prove results about them analogous to well known classical facts. After these preliminary matters are settled, we will closely follow [5] to obtain the notion of $\mathbf{\Sigma}^0_n$-effective measurability and the analogue of Brattka's Completeness Theorem for computable $\mathbb{R}_\kappa$-metric spaces.

2.1 Representations for $\mathbb{R}_\kappa$-metric spaces

We introduce representations for κ-separable $\mathbb{R}_\kappa$-metric spaces which we will use in the rest of the thesis and we prove some of their properties.

Definition 2.1.1 (Computable κ-metric spaces)**.** *A computable κ-metric space $\mathfrak{X}$ is a triple (X, d, s) where (X, d) is a $\mathbb{R}_\kappa$-metric space, $s\colon \kappa \to X$ is a sequence which is dense in $\mathfrak{X}$, and the relations $L_\mathfrak{X}, R_\mathfrak{X} \subseteq \kappa^3$ given by*

$$(\alpha,\beta,\gamma) \in L_\mathfrak{X} \iff d(s(\alpha), s(\beta)) < \overline{\gamma},$$

$$(\alpha,\beta,\gamma) \in R_\mathfrak{X} \iff d(s(\alpha), s(\beta)) > \overline{\gamma},$$

are semidecidable by a Type 1 κ-Turing machine.

In other words, the last condition implies that for every α, β and every γ, the problems of determining whether $s(\alpha) \in B_\mathfrak{X}(s(\beta), \overline{\gamma})$ or whether $s(\alpha) \notin \overline{B}_\mathfrak{X}(s(\beta), \overline{\gamma})$ are semidecidable by a κ-Turing Machine.

Proposition 2.1.2. *Assuming Hypothesis 1.2.19, the following are computable κ-metric spaces:*

(a) $(\kappa, d_\kappa, \mathrm{id}_\kappa)$ *where d_κ is the discrete metric and id_κ is the identity function,*

(b) $(\kappa^\kappa, d_{\kappa^\kappa}, \mu')$ *where $\mu'\colon \kappa \to \kappa^\kappa$ is given by $\mu'(\alpha) = \mu(\alpha)0^\kappa$ and d_{κ^κ} is given by, for all $p \neq q \in \kappa^\kappa$*

$$d_{\kappa^\kappa}(p, q) = \tfrac{1}{\alpha+1},$$

where $\alpha = \min\{\beta \in \kappa \mid p(\beta) \neq q(\beta)\}$,

(c) $(2^\kappa, d_{2^\kappa}, \nu')$ *where $\nu'\colon \kappa \to 2^\kappa$ is given by $\nu'(\alpha) = \nu(\alpha)0^\kappa$ and d_{2^κ} is given by, for all $p \neq q \in 2^\kappa$*

$$d_{2^\kappa}(p, q) = \tfrac{1}{\alpha+1},$$

where $\alpha = \min\{\beta \in \kappa \mid p(\beta) \neq q(\beta)\}$, *(d)* $(\mathbb{R}_\kappa, d_{\mathbb{R}_\kappa}, \nu_{\mathbb{Q}_\kappa})$ *where* $d_{\mathbb{R}_\kappa}$ *is given by*

$$d_{\mathbb{R}_\kappa}(x,y) = |x-y|,$$

where $|\cdot|$ *and* $-$ *are the operations defined on* $\mathbb{R}_\kappa$*.*

Proof. The claim is clearly true for $(\kappa, d_\kappa, \mathrm{id}_\kappa)$. For $(\kappa^\kappa, d_{\kappa^\kappa}, \mu')$, the fact that d_{κ^κ} is an $\mathbb{R}_\kappa$-metric is obvious by definition (note that this metric induces the standard bounded topology). The fact that $\mathrm{ran}(\mu')$ is dense in κ^κ is also a direct consequence of the definition. To see that the relations L_{κ^κ} and R_{κ^κ} are semidecidable note that μ' is computable by our discussion in Section 1.2.3 and finding the least ordinal where two short sequences differ is clearly a κ-computable operation. The proof for $(2^\kappa, d_{2^\kappa}, \nu')$ is analogous. For $(\mathbb{R}_\kappa, d_{\mathbb{R}_\kappa}, \nu_{\mathbb{Q}_\kappa})$: $\mathrm{ran}(\nu_{\mathbb{Q}_\kappa}) = \mathbb{Q}_\kappa$ is dense in $\mathbb{R}_\kappa$ by definition, and the semidecidability of $L_{\mathbb{R}^\kappa}$ and $R_{\mathbb{R}^\kappa}$ follows immediately from the computability of $\nu_{\mathbb{Q}_\kappa}$ and Proposition 1.4.33. □

Given a computable κ-metric space $\mathfrak{X} = (X, d, s)$, we obtain a representation of $\mathfrak{X}$ as follows: define $\delta_{\mathfrak{X}} :\subseteq \kappa^\kappa \to X$ as

$$\delta_{\mathfrak{X}}(p) = \lim_{\alpha\to\kappa} s(p(\alpha))$$

if $\lim_{\alpha\in\kappa} s(p(\alpha))$ exists and the sequence $(s(p(\alpha)))_{\alpha\in\kappa}$ is a *fast convergent Cauchy sequence*, i.e. for all $\alpha < \beta$, we have

$$d\big(s(p(\alpha)), s(p(\beta))\big) < \tfrac{1}{\alpha+1}. \tag{2.1}$$

This is called the *Cauchy representation* of (X, d, s). In the rest of this thesis, unless explicitly stated, whenever we consider functions between κ-metric spaces, we will assume that these are equipped with their Cauchy representations. We will refer to condition (2.1) as the *fast convergence* condition for sequences/codes. The results in Chapter 4 of [16] immediately imply that the Cauchy representation so obtained is admissible with respect to the metric topology on X (see [41], Theorem 8.1.4 for the result in the classical context). Moreover, we know from [16] Lemmas 4.3.1 and 4.3.3 that products and subspace representations obtained from admissible representations are in turn admissible. This allows us to apply the Main Theorem of Generalized Computable Analysis and identify continuous partial functions with partial functions admitting a continuous realizer whenever we reason with functions on computable κ-metric spaces and spaces built from them. We will often do this without mention.

We introduce operations to obtain new computable metric spaces from old ones:

Lemma 2.1.3 (Subspaces and product spaces)**.** *Let* $\mathfrak{X} = (X, d, s)$ *and* $\mathfrak{Y} = (Y, d', s')$ *be computable* κ*-metric spaces:*

(a) for any $A \subseteq X$*, the triple* $(A, d\restriction(A\times A), s\restriction A)$ *is a computable* κ*-metric space,*

(b) the space $\mathfrak{X}\times\mathfrak{Y} = (X\times Y, d'', s'')$ *where*

$$s''(\ulcorner\alpha,\beta\urcorner) = (s(\alpha), s'(\beta))$$

and

$$d''((x,y),(x',y')) = \max\{d(x,x'), d'(y,y')\}$$

is a computable κ*-metric space.*

Proof. Both cases are proved rather straightforwardly from the definition, we give the proof for the product space $\mathfrak{X}\times\mathfrak{Y}$: it is obvious that $d_{X\times Y}$ makes $(X\times Y, d_{X\times Y})$ into an $\mathbb{R}_\kappa$-metric space. To see that $\mathrm{ran}(s'')$ is a dense subset of $X\times Y$, let B be a basic open ball in $\mathfrak{X}\times\mathfrak{Y}$, by definition there must be $x\in X, y\in Y$ and $q\in\mathbb{Q}_\kappa$ such that $B_{\mathfrak{X}}(x,q)\times B_{\mathfrak{Y}}(y,q)\subseteq B$. By density of $\mathrm{ran}(s)\in\mathfrak{X}$ and $\mathrm{ran}(s')\in\mathfrak{Y}$, let α and β be such that $s(\alpha)\in B_{\mathfrak{X}}(x,q)$ and $s'(\beta)\in B_{\mathfrak{Y}}(y,q)$, then $s''(\ulcorner\alpha,\beta\urcorner)\in B$, proving density of s'' in $\mathfrak{X}\times\mathfrak{Y}$. Deciding whether a triple (α,β,γ) belongs to $L_{\mathfrak{X}\times\mathfrak{Y}}$ is done by checking whether the appropriately corresponding ordinals (α',β',γ') and $(\alpha'',\beta'',\gamma'')$ belong to $L_{\mathfrak{X}}$ and $L_{\mathfrak{Y}}$ respectively. The case for $R_{\mathfrak{X}\times\mathfrak{Y}}$ is analogous. □

2.1.1 Technical results on Cauchy representations

We prove some results on Cauchy representations which provide insight on how we can recursively build names for points in a computable κ-metric space. These results will be needed in later constructions, primarily in the proof of Theorem 2.2.2.

Lemma 2.1.4. *Let $\mathfrak{X} = (X, d, s)$ be a computable κ-metric space and let $p \in \mathrm{dom}(\delta_{\mathfrak{X}})$, $\alpha \in \kappa$ and $x \in X$. If $\delta_{\mathfrak{X}}(p) = x$, then*

$$x \in \mathrm{cl}(B_{\mathfrak{X}}(s(p(\alpha)), \tfrac{1}{\alpha+1})),$$

in particular $x \in \overline{B}_{\mathfrak{X}}(s(p(\alpha)), \frac{1}{\alpha+1})$.

Proof. First, if $p \in \mathrm{dom}(\delta_{\mathfrak{X}})$, the fast convergence condition imposes that for every $\beta > \alpha$,

$$s(p(\beta)) \in B_{\mathfrak{X}}(s(p(\alpha)), \tfrac{1}{\alpha+1}).$$

Hence $s(p(\beta))_{\alpha \leq \beta < \kappa}$ is a sequence fully contained in $B_{\mathfrak{X}}(s(p(\alpha)), \frac{1}{\alpha+1})$ which converges to x, so clearly $x \in \mathrm{cl}(B_{\mathfrak{X}}(s(p(\alpha)), \frac{1}{\alpha+1}))$. The fact that x belongs to the closed ball follows because in any $\mathbb{R}_\kappa$-metric space, for all points y and radii r,

$$\mathrm{cl}(B(y, r)) \subseteq \overline{B}(y, r).$$

□

Definition 2.1.5 (Compatible points)**.** *Let $\mathfrak{X} = (X, d, s)$ be a computable κ-metric space and let $w \in \kappa^{<\kappa}$ be a word. We define the set of points* compatible with *w as*

$$\mathrm{compat}_w = \bigcap_{\alpha \in \mathrm{dom}(w)} B_{\mathfrak{X}}(s(w(\alpha)), \tfrac{1}{\alpha+1}).$$

Notice that for every $w \in \kappa^{<\kappa}$, the set compat_w is open as it is the intersection of fewer than κ open balls and any computable κ-metric space is κ-additive by Proposition 1.1.7. Moreover, the sets compat_w allow us to state a refinement of Lemma 2.1.4.

Lemma 2.1.6. *Let $\mathfrak{X} = (X, d, s)$ be a computable κ-metric space and let $p \in \mathrm{dom}(\delta_{\mathfrak{X}})$, $\alpha \in \kappa$ and $x \in X$. If $\delta_{\mathfrak{X}}(p) = x$, and $w = p{\restriction}\alpha$, then $x \in \mathrm{cl}(\mathrm{compat}_w)$.*

Proof. Similarly to the proof of Lemma 2.1.4, the fast convergence condition on $s(p(\gamma))_{\gamma \in \kappa}$ implies that $s(p(\beta)) \in \mathrm{compat}_w$ for all $\beta \geq \alpha$, hence $x \in \mathrm{cl}(\mathrm{compat}_w)$. □

Notice in particular that if $w = p{\restriction}\alpha$ for some $p \in \mathrm{dom}(\delta_{\mathfrak{X}})$ and $\alpha \in \kappa$, then it must be the case that $\mathrm{compat}_w \neq \emptyset$.

Definition 2.1.7 (Extendible words)**.** *Given a computable κ-metric space $\mathfrak{X} = (X, d, s)$, we define the tree of extendible words $w \in \kappa^{<\kappa}$ given by*

$$\mathrm{extw}_{\mathfrak{X}} = \{w \in \kappa^{<\kappa} \mid \exists p \in \mathrm{dom}(\delta_{\mathfrak{X}})\ (w \subseteq p)\}.$$

We obtain a partial converse to Lemma 2.1.6:

Lemma 2.1.8. *Let $\mathfrak{X}$ be a computable κ-metric space and let $w \in \kappa^{<\kappa}$ be such that* $\mathrm{compat}_w \neq \emptyset$, *then $w \in \mathrm{extw}_{\mathfrak{X}}$. More precisely, for every $x \in \mathrm{compat}_w$ there exists $p_x \in \mathrm{dom}(\delta_{\mathfrak{X}})$ such that $w \subseteq p_x$ and $\delta_{\mathfrak{X}}(p_x) = x$.*

Proof. It is obvious that the first claim follows from the second, therefore we just need to prove that every $x \in \mathrm{compat}_w$ admits a name p_x which extends w. To do so, let

$$\lambda = \min\left\{\beta \in \kappa \mid B_{\mathfrak{X}}(x, \tfrac{1}{\lambda+1}) \subseteq \mathrm{compat}_w\right\}.$$

Note that the set of ordinals on the right hand side of the definition is nonempty because compat_w is an open neighbourhood of x and the sequence $(\frac{1}{\lambda+1})_{\lambda\in\kappa}$ is coinitial in $\mathbb{R}_\kappa$. Let $\lambda' = \max\{\lambda, |w|\}$ and consider the sequence of open balls $(B_\eta)_{\lambda'\leq\eta<\kappa}$ given by

$$B_\eta = B_{\mathfrak{X}}(x, \tfrac{1}{(\eta+1)\cdot 3}).$$

Notice that, since $\eta \geq \lambda' \geq \lambda$, it follows that each B_η is contained in compat_w. By density of $\mathrm{ran}(s)$ in $\mathfrak{X}$, it follows that there is a sequence of ordinals $p \in \kappa^\kappa$ such that $s(p(\alpha)) \in B_{\eta+\alpha}$ for all $\alpha \in \kappa$. Now let $\alpha < \beta \in \kappa$, by construction $s(p(\beta)) \in B_{\eta+\beta} \subseteq B_{\eta+\alpha}$, hence $s(p(\beta)), s(p(\alpha))$ both belong to $B_{\eta+\alpha}$. Since the latter is an open ball with radius $\frac{1}{(\eta+\alpha+1)\cdot 3}$, it follows that

$$d(s(p(\beta), s(p(\alpha)))) \leq \tfrac{2}{(\eta+\alpha+1)\cdot 3} < \tfrac{1}{\eta+\alpha+1} \leq \tfrac{1}{|w|+\alpha+1}.$$

In particular this implies that the sequence determined by the word $p_x = w^\frown p$ satisfies condition (2.1). By construction the limit of the sequence $s(p_x(\alpha))_{\alpha\in\kappa}$ is x, hence $p_x \in \mathrm{dom}(\delta_{\mathfrak{X}})$ and $\delta_{\mathfrak{X}}(p_x) = x$, proving the claim. $\square$

Given any point x of a computable κ-metric space $\mathfrak{X}$, we can introduce a set of nice partial names for x. These are extendible words in κ^κ which x is compatible with.

Definition 2.1.9 (Nice partial names)**.** *Let $\mathfrak{X} = (X, d, s)$ be a computable κ-metric space and let x be an element of X, define the set of its nice partial names as $\mathrm{npn}_x = \{w \in \mathrm{extw}_{\mathfrak{X}} \mid x \in \mathrm{compat}_w\}$.*

It is clear that npn_x is a subtree of $\mathrm{extw}_{\mathfrak{X}}$, and that a branch in npn_x corresponds to a sequence p such that $\delta_{\mathfrak{X}}(p) = x$, although it is not immediately obvious from the definition that npn_x has branches at all. We prove that this is always the case, moreover we show that each npn_x is $<\kappa$-closed and $\mathrm{ext}(\mathrm{npn}_x) = \mathrm{npn}_x$. These properties will be instrumental in the proof of Theorem 2.2.2.

Lemma 2.1.10. *Let $\mathfrak{X} = (X, d, s)$ be a computable κ-metric space and let $x \in X$. The tree npn_x is $<\kappa$-closed and satisfies $\mathrm{ext}(\mathrm{npn}_x) = \mathrm{npn}_x$, in particular $[\mathrm{npn}_x] \neq \emptyset$.*

Proof. We first show $<\kappa$-closedness by contraposition, i.e. we show that if a word w of limit length is not in npn_x, there must be a proper initial segment $w' \subseteq w$ such that $w' \notin w$. To this end, let λ be any limit ordinal and let $w \in \mathrm{extw}_{\mathfrak{X}}$ be a word with $|w| = \lambda$. Assume that $w \notin \mathrm{npn}_x$, then $x \notin \mathrm{compat}_w$, i.e.

$$x \notin \bigcap_{\alpha\in\lambda} B_{\mathfrak{X}}(s(w(\alpha)), \tfrac{1}{\alpha+1}).$$

Trivially this implies that there is some $\beta \in \lambda$ such that

$$x \notin B_{\mathfrak{X}}(s(w(\beta)), \tfrac{1}{\beta+1}),$$

hence $w\restriction(\beta+1) \notin \mathrm{npn}_x$.

Now we show that $\mathrm{ext}(\mathrm{npn}_x) = \mathrm{npn}_x$, which implies in particular that $[\mathrm{npn}_x] \neq \emptyset$ as npn_x is obviously nonempty. To this end, let $w \in \mathrm{npn}_x$, then by definition $x \in \mathrm{compat}_w$ and $w \in \mathrm{extw}_{\mathfrak{X}}$, therefore we can consider the sequence $p \in \kappa^\kappa$ defined in the proof of Lemma 2.1.8 such that $w^\frown p$ is a $\delta_{\mathfrak{X}}$-name of x. Notice that for every $\alpha \in \kappa$, $p(\alpha) \in B_{\eta+\alpha}$ where $\eta \geq |w|$ and

$$B_{\eta+\alpha} = B_{\mathfrak{X}}(x, \tfrac{1}{(\eta+\alpha+1)\cdot 3}),$$

hence in particular $d(x, s(p(\alpha))) < \frac{1}{|w|+\alpha+1}$ and consequently for every β,

$$x \in B_{\mathfrak{X}}(s(w^\frown p(\beta)), \tfrac{1}{\beta+1}).$$

In other words, if we let $w_\beta = (w^\frown p)\restriction\beta$, this implies that for all β, $x \in \mathrm{compat}_{w_\beta}$. This implies that $w^\frown p$ is a sequence such that each of its initial segments belongs to npn_x, hence a branch in npn_x. In particular this means that $w \in \mathrm{ext}(\mathrm{npn}_x)$ and thus proves that $\mathrm{ext}(\mathrm{npn}_x) = \mathrm{npn}_x$. $\square$

2.1.2 Representation for the κ-Borel sets in computable metric spaces

We now turn to the definition of representations for the finite levels of the κ-Borel hierarchy of a computable metric space (X, d, s) (cf. Definition 1.1.24) and we prove a proposition which is the generalized analogue of [5, Proposition 3.2]. We will repeatedly use this result when working with represented spaces of Borel sets.

Definition 2.1.11 (Representation of finite Borel levels)**.** *Let $\mathfrak{X} = (X, d, s)$ be a computable κ-metric space. Define the following representations:*

- $\delta_{\boldsymbol{\Sigma}^0_1(\mathfrak{X})} :\subseteq \kappa^\kappa \to \boldsymbol{\Sigma}^0_1(\mathfrak{X})$ *given by*

$$\delta_{\boldsymbol{\Sigma}^0_1(\mathfrak{X})}(p) = U \iff U = \bigcup\{B(s(\alpha, \overline{\beta})) \mid \ulcorner\alpha, \beta\urcorner \in \operatorname{ran}(p)\},$$

- *for all $n \in \omega$,* $\delta_{\boldsymbol{\Pi}^0_n(\mathfrak{X})} :\subseteq \kappa^\kappa \to \boldsymbol{\Pi}^0_n(\mathfrak{X})$ *given by*

$$\delta_{\boldsymbol{\Pi}^0_n(\mathfrak{X})}(p) = C \iff \delta_{\boldsymbol{\Sigma}^0_n(\mathfrak{X})}(p) = X \setminus C,$$

- *for all $n \in \omega$,* $\delta_{\boldsymbol{\Sigma}^0_{n+1}(\mathfrak{X})} :\subseteq \kappa^\kappa \to \boldsymbol{\Sigma}^0_{n+1}(\mathfrak{X})$ *given by*

$$\delta_{\boldsymbol{\Sigma}^0_{n+1}(\mathfrak{X})}(p) = U \iff U = \bigcup_{\alpha\in\kappa} \delta_{\boldsymbol{\Pi}^0_n(\mathfrak{X})}(p_\alpha),$$

- *for all $n \in \omega$,* $\delta_{\boldsymbol{\Delta}^0_n(\mathfrak{X})} :\subseteq \kappa^\kappa \to \boldsymbol{\Delta}^0_n(\mathfrak{X})$ *given by*

$$\delta_{\boldsymbol{\Delta}^0_n(\mathfrak{X})}(p) = D \iff p = \langle r, q\rangle,$$

  with $\delta_{\boldsymbol{\Sigma}^0_n(\mathfrak{X})}(r) = D$ *and* $\delta_{\boldsymbol{\Pi}^0_n(\mathfrak{X})}(q) = D$.

Proposition 2.1.12. *The following operations are computable, for all $n \in \omega$ and all computable κ-metric spaces $\mathfrak{X} = (X, d, s)$ and $\mathfrak{Y} = (Y, d', s')$:*

(a) the complementation maps $\boldsymbol{\Sigma}^0_n \to \boldsymbol{\Pi}^0_n$, $\boldsymbol{\Pi}^0_n \to \boldsymbol{\Sigma}^0_n$, $\boldsymbol{\Delta}^0_n \to \boldsymbol{\Delta}^0_n$ all given by $A \mapsto X \setminus A$,

(b) the κ-union maps $(\boldsymbol{\Sigma}^0_n)^\kappa \to \boldsymbol{\Sigma}^0_n$ given by $(A_\alpha)_{\alpha\in\kappa} \mapsto \bigcup_{\alpha\in\kappa} A_\alpha$,

(c) the κ-intersection maps $(\boldsymbol{\Pi}^0_n)^\kappa \to \boldsymbol{\Pi}^0_n$ given by $(A_\alpha)_{\alpha\in\kappa} \mapsto \bigcap_{\alpha\in\kappa} A_\alpha$,

(d) the inclusion maps $\boldsymbol{\Sigma}^0_n \hookrightarrow \boldsymbol{\Sigma}^0_{n+1}$, $\boldsymbol{\Sigma}^0_n \hookrightarrow \boldsymbol{\Pi}^0_{n+1}$, $\boldsymbol{\Pi}^0_n \hookrightarrow \boldsymbol{\Pi}^0_{n+1}$, $\boldsymbol{\Pi}^0_n \hookrightarrow \boldsymbol{\Sigma}^0_{n+1}$, $\boldsymbol{\Delta}^0_n \hookrightarrow \boldsymbol{\Sigma}^0_n$, $\boldsymbol{\Delta}^0_n \hookrightarrow \boldsymbol{\Pi}^0_n$ all given by $A \mapsto A$,

(e) assuming Hypothesis 1.2.19, the $<\kappa$-intersection maps $(\boldsymbol{\Sigma}^0_n)^\kappa \times \kappa \to \boldsymbol{\Sigma}^0_n$ given by $((A_\alpha)_{\alpha\in\kappa}, \lambda) \mapsto \bigcap_{\alpha\in\lambda} A_\alpha$ and the $<\kappa$-union maps $(\boldsymbol{\Pi}^0_n)^\kappa \times \kappa \to \boldsymbol{\Pi}^0_n$ given by $((A_\alpha)_{\alpha\in\kappa}, \lambda) \mapsto \bigcup_{\alpha\in\lambda} A_\alpha$,

(f) the product maps $\boldsymbol{\Sigma}^0_n(\mathfrak{X}) \times \boldsymbol{\Sigma}^0_n(\mathfrak{Y}) \to \boldsymbol{\Sigma}^0_n(\mathfrak{X} \times \mathfrak{Y})$ and $\boldsymbol{\Pi}^0_n(\mathfrak{X}) \times \boldsymbol{\Pi}^0_n(\mathfrak{Y}) \to \boldsymbol{\Pi}^0_n(\mathfrak{X} \times \mathfrak{Y})$ given by

$$(A, B) \mapsto A \times B.$$

We assume that all spaces mentioned are equipped with the representations given in Definition 2.1.11, accordingly we assume that sequence spaces and product spaces are equipped with the corresponding sequence and product representations.

Proof. The computability of the complement operations $\boldsymbol{\Sigma}^0_n \to \boldsymbol{\Pi}^0_n$ follows directly from the definitions as the identity on generalized Baire space is a realizer for it. For the complement operations $\boldsymbol{\Delta}^0_n \to \boldsymbol{\Delta}^0_n$, observe that if A has $\boldsymbol{\Delta}^0_n$-name given by $\langle p, q\rangle$, then a $\boldsymbol{\Delta}^0_n$-name of A^{c} is given by $\langle q, p\rangle$ and the function $\langle p, q\rangle \mapsto \langle q, p\rangle$ is obviously computable.

The computability of the κ-union maps also follows from definitions in the case $n = 1$: a $(\boldsymbol{\Sigma}^0_1)^\kappa$-name for $(A_\alpha)_{\alpha\in\kappa}$ is just a sequence $p \in \kappa^\kappa$ where each p_α is a $\boldsymbol{\Sigma}^0_1$-name for A_α hence an enumeration of (codes for) open balls exhausting A_α. It is clear that then p is also a $\mathbf{\Sigma}^0_1$-name for $\bigcup_{\alpha\in\kappa} A_\alpha$. For the general case the reasoning is analogous but a bit more involved, for any $n \in \omega$, a $(\mathbf{\Sigma}^0_{n+1})^\kappa$-name of a sequence $(A_\alpha)_{\alpha\in\kappa}$ is a sequence p where each p_α is a $\mathbf{\Sigma}^0_{n+1}$-name of A_α, so for every $\delta \in \kappa$, $(p_\alpha)_\delta$ is a $\mathbf{\Pi}^0_n$-name of a set C^α_δ such that $A_\alpha = \bigcup_{\delta\in\kappa} C^\alpha_\delta$, hence $\bigcup_{\alpha\in\kappa} A_\alpha = \bigcup_{\ulcorner\alpha,\delta\urcorner\in\kappa} C^\alpha_\delta$. We can use the pairing function to compute from p a sequence p' such that $p'_{\ulcorner\alpha,\delta\urcorner} = (p_\alpha)_\delta$. This p' is a $\mathbf{\Sigma}^0_{n+1}$-name for $\bigcup_{\alpha\in\kappa} A_\alpha$. Since this procedure is uniform in the code p, this shows that the κ-union maps are computable.

The computability of the κ-intersection maps follows directly from the definitions and the computability of the union maps: a $(\mathbf{\Pi}^0_n)^\kappa$-name for $(A_\alpha)_{\alpha\in\kappa}$ is also a $(\mathbf{\Sigma}^0_n)^\kappa$-name for $(A^{\mathrm{c}}_\alpha)_{\alpha\in\kappa}$, so it can be used to compute a $\mathbf{\Sigma}^0_n$-name for $\bigcup_{\alpha\in\kappa} A^{\mathrm{c}}_\alpha$, which is the same as a $\mathbf{\Pi}^0_n$-name for its complement $\bigcap_{\alpha\in\kappa} A_\alpha$.

For the inclusion maps, first note that the inclusions $\mathbf{\Delta}^0_n \hookrightarrow \mathbf{\Sigma}^0_n$ and $\mathbf{\Delta}^0_n \hookrightarrow \mathbf{\Pi}^0_n$ are obviously computable by definition, as is the inclusion $\mathbf{\Pi}^0_n \hookrightarrow \mathbf{\Sigma}^0_{n+1}$. The computability of the inclusions $\mathbf{\Pi}^0_n \hookrightarrow \mathbf{\Pi}^0_{n+1}$ and $\mathbf{\Sigma}^0_n \hookrightarrow \mathbf{\Pi}^0_{n+1}$ follows from the computability of the corresponding inclusions with the roles of $\mathbf{\Pi}$ and $\mathbf{\Sigma}$ exchanged and the computability of the complementation maps. Therefore all we have to prove is the computability of $\mathbf{\Sigma}^0_n \hookrightarrow \mathbf{\Sigma}^0_{n+1}$ for all $n \in \omega$.

We prove this by induction on n: for $n = 1$, a $\mathbf{\Sigma}^0_1$-name for a set A is a sequence p such that $A = \bigcup_{\ulcorner\alpha,\beta\urcorner\in\operatorname{ran}(p)} B(s(\alpha), \overline{\beta})$. Now, for every (α, β), we computably obtain a $\mathbf{\Pi}^0_1$ name of $\overline{B}(s(\alpha), \overline{\beta})$ as

$$(\overline{B}(s(\alpha), \overline{\beta}))^{\mathrm{c}} = \bigcup\{B(s(\eta), \overline{\gamma}) \mid d(s(\alpha), s(\eta)) > \overline{\beta} + \overline{\gamma}\}.$$

Now note that

$$B(s(\delta), \overline{\xi}) = \bigcup\{\overline{B}(s(\iota), \overline{\lambda}) \mid d(s(\delta), s(\iota)) < \overline{\xi} - \overline{\lambda}\}.$$

Therefore we can computably obtain a $\mathbf{\Sigma}^0_2$-name of every open ball in the name of A. This, together with the computability of the union maps establishes the claim for $n = 1$.

For the inductive step assume that we know that $\mathbf{\Sigma}^0_n \hookrightarrow \mathbf{\Sigma}^0_{n+1}$ and $\mathbf{\Pi}^0_n \hookrightarrow \mathbf{\Pi}^0_{n+1}$ are computable (note that by computability of the complementation maps, one of these two implies the other) and let p be a $\mathbf{\Sigma}^0_{n+1}$-name for a set A: then each p_α is a $\mathbf{\Pi}^0_n$-name for some C_α such that $\bigcup_{\alpha\in\kappa} C_\alpha = A$. By the inductive assumption, we can compute (uniformly in α) a $\mathbf{\Pi}^0_{n+1}$-name p'_α for each C_α and obtain the $\mathbf{\Sigma}^0_{n+2}$-name for A given by $\langle(p'_\alpha)_{\alpha\in\kappa}\rangle$. Again this procedure is uniform in p, therefore the inclusion $\mathbf{\Sigma}^0_{n+1} \hookrightarrow \mathbf{\Sigma}^0_{n+2}$ is computable.

For the $<\kappa$-union and $<\kappa$-intersection maps, first note that they are well defined by Proposition 1.1.25. We prove their computability by induction on n: for $n = 1$, let $(A_\alpha)_{\alpha\in\kappa} \in (\mathbf{\Sigma}^0_1)^\kappa$, let $\lambda \in \kappa$ and let $A_\alpha = \bigcup_{\beta\in\kappa} B^\alpha_\beta$ where for every $\beta \in \kappa$, B^α_β is a basic open ball. As in the proof of Proposition 1.1.25, for every $f \in \kappa^\lambda$, we define $B^\alpha_f = B^\alpha_{f(\alpha)}$ and obtain that

$$\bigcap_{\alpha\in\lambda} A_\alpha = \bigcap_{\alpha\in\lambda}\bigcup_{\beta\in\kappa} B^\alpha_\beta = \bigcup_{f\in\kappa^\lambda}\bigcap_{\alpha\in\lambda} B^\alpha_f.$$

Now note that for every $f \in \kappa^\lambda$, a name for $\bigcap_{\alpha\in\lambda} B^\alpha_f$ can be computed as follows: run through the ordinals $<\kappa$ and for each $\delta = \ulcorner\beta, \gamma\urcorner$, check whether $B(s(\beta), \overline{\gamma}) \subseteq B^\alpha_f$ for all $\alpha \in \lambda$. If so, enumerate the ball corresponding to δ in the name for $\bigcap_{\alpha\in\lambda} B^\alpha_f$ (the problem of determining whether $B(s(\beta), \overline{\gamma}) \subseteq B^\alpha_f$ is semidecidable by definition of κ-metric space). Doing this for every $\delta \in \kappa$ yields a $\mathbf{\Sigma}^0_1$-name of $\bigcap_{\alpha\in\lambda} B^\alpha_f$ in the long run. Therefore a machine which is given λ and a name for $(A_\alpha)_{\alpha\in\kappa}$ can output a name for $\bigcap_{\alpha\in\lambda} A_\alpha$ by simply generating functions $f \in \kappa^\lambda$ and for each of them computing $\bigcap_{\alpha\in\lambda} B^\alpha_f$. Computability of small unions of $\mathbf{\Pi}^0_1$ sets now follows by considering complements, exploiting the definition of $\mathbf{\Pi}^0_1$-names.

Now assume that the claim holds for the level n, and let p be a $(\mathbf{\Sigma}^0_{n+1})^\kappa$-name for $(A_\alpha)_{\alpha\in\kappa}$, so for every α we have access to a $(\mathbf{\Pi}^0_n)^\kappa$-name of $(C^\alpha_\beta)_{\beta\in\kappa}$ such that $A_\alpha = \bigcup_{\beta\in\kappa} C^\alpha_\beta$.
Let $\lambda \in \kappa$, again we have

$$\bigcap_{\alpha\in\lambda} A_\alpha = \bigcap_{\alpha\in\lambda}\bigcup_{\beta\in\kappa} C^\alpha_\beta = \bigcup_{f\in\kappa^\lambda}\bigcap_{\alpha\in\lambda} C^\alpha_f,$$

and by the inductive assumption we can (uniformly in α) compute a name $\mathbf{\Pi}^0_n$-name for $\bigcap_{\alpha\in\lambda} C^\alpha_f$ for every $f \in \kappa^\lambda$. Interleaving these names gives a name for $\bigcap_{\alpha\in\lambda} A_\alpha$. Computability of small intersection of $\mathbf{\Pi}^0_{n+1}$ sets again follows directly. This concludes the induction and proves the claim.

Lastly for the product operation, we again prove the claim by induction on n: let s'' and d'' be as in the definition of the product metric space $\mathfrak{X} \times \mathfrak{Y} = (X \times Y, d'', s'')$, then a basic open set $B_{\mathfrak{X}}(s(\alpha), \overline{\beta}) \times B_{\mathfrak{Y}}(s'(\gamma), \overline{\delta})$ can be written as:

$$\bigcup\{B_{\mathfrak{X}\times\mathfrak{Y}}(s''(\ulcorner\xi_1, \xi_2\urcorner), \overline{\eta}) \mid d(s(\xi_1), s(\alpha)) < \overline{\beta} - \overline{\eta} \wedge d'(s'(\xi_2), s'(\gamma)) < \overline{\delta} - \overline{\eta}\}.$$

A $\mathbf{\Sigma}^0_1(\mathfrak{X})$-name of some open $A \subseteq X$ is a sequence of codes of basic open balls B_α with $\bigcup_{\alpha\in\kappa} B_\alpha = A$, similarly a $\mathbf{\Sigma}^0_1(\mathfrak{Y})$-name for some $C \subseteq Y$ is a sequence of D_α with the analogous property. Notice that

$$A \times C = \bigcup_{\alpha\in\kappa} B_\alpha \times \bigcup_{\alpha\in\kappa} D_\alpha = \bigcup_{\ulcorner\alpha,\beta\urcorner\in\kappa} A_\alpha \times D_\beta,$$

and we can compute $\mathbf{\Sigma}^0_1$-names for each element in the union, hence we can compute a $\mathbf{\Sigma}^0_1$-name for $A \times C$. The result for closed sets follows immediately, given an closed name for $A \subseteq X$ and a closed name for $B \subseteq Y$, we have open names for A^c and B^c, which can be used to compute open names for $A^c \times Y$ and $X \times B^c$, which in turn are closed names for $A \times Y$ and $X \times B$. Since $(A \times Y) \cap (X \times B) = A \times B$, we obtain the computability for closed names of products of closed sets. Now the result for the remaining finite levels of the hierarchy follows by an induction similar to the ones above. □

2.2 $\mathbf{\Sigma}^0_n$-effective computability

We now introduce the concept of $\mathbf{\Sigma}^0_n$-effective measurability, which is the computable counterpart of $\mathbf{\Sigma}^0_n$-measurability for functions on $\mathbb{R}_\kappa$-metric spaces. Both notions are analogous to the corresponding notions for ordinary metric spaces. The main result of this section is Theorem 2.2.2, which is the generalization of a well known result in the theory of computable metric spaces (see [4, Theorem 6.2]). Moreover, we prove results on effective measurability which are needed to prove Proposition 2.3.7 (the closure of the classes of $\mathbf{\Sigma}^0_n$-computable functions under Weihrauch reductions), which is in turn a necessary ingredient for our Completeness Theorem (Theorem 2.3.12) in the next section.

Definition 2.2.1 (Effective measurability)**.** *For any $n \in \omega$, a $\mathbf{\Sigma}^0_n$-measurable multi-valued operation between computable κ-metric spaces $f :\subseteq \mathfrak{X} \rightrightarrows \mathfrak{Y}$ is called* $\mathbf{\Sigma}^0_n$-effectively measurable *(or* computable*) if there exists a $(\delta_{\mathbf{\Sigma}^0_1(\mathfrak{Y})}, \delta_{\mathbf{\Sigma}^0_n(\mathfrak{X})})$-computable $\Phi \colon \mathbf{\Sigma}^0_1(\mathfrak{Y}) \rightrightarrows \mathbf{\Sigma}^0_n(\mathfrak{X})$ such that $f^{-1}[U] = V \cap \mathrm{dom}(f)$ for all $V \in \Phi(U)$.*

A map between metric spaces is continuous if and only if it is $\mathbf{\Sigma}^0_1$-measurable. A similar result holds for computable maps, giving credit to the idea that $\mathbf{\Sigma}^0_n$-effective measurability is a sensible relaxation of the notion of computability.

Theorem 2.2.2. *Let $\mathfrak{X} = (X, d, s)$ and $\mathfrak{Y} = (Y, d', s')$ be computable κ-metric spaces and let $f :\subseteq \mathfrak{X} \to \mathfrak{Y}$ be a partial function. If f is $\mathbf{\Sigma}^0_1$-computable, then it is computable. Assuming Hypothesis 1.2.19, the converse holds as well.*

Proof. Assume that f is computable and let F be one of its computable realizers, we want to show that there is an operation $\Phi \colon \mathbf{\Sigma}^0_1(\mathfrak{Y}) \rightrightarrows \mathbf{\Sigma}^0_1(\mathfrak{X})$ such that for all $U \in \mathbf{\Sigma}^0_1(\mathfrak{Y})$ and all $V \in \Phi(U)$, $f^{-1}[U] = V \cap \mathrm{dom}(f)$ and that Φ has a computable realizer G. We describe the behaviour of a machine computing G on basic open balls: let $B = B_{\mathfrak{Y}}(s'(\alpha), \overline{\beta})$. For every $w \in \kappa^{<\kappa}$, consider words of the form $w\eta^\kappa$ and for each of them run F in parallel[1] until (if ever) the output tape meets the following conditions:[2] if $t \in \kappa^{<\kappa}$ is the portion of output printed so far, then $|t| = \lambda + 1$ for some λ

[1] This is where the assumption of computable enumerability of $2^{<\kappa}/\kappa^\kappa$ comes into play in the proof.

[2] Notice that by definition of computable κ-metric space, a κ-Turing Machine can compute whether $d_{\mathfrak{Y}}(s'(\alpha), s'(t(\lambda))) < \overline{\beta} - \frac{2}{\lambda+1}$. In our algorithm, the machine performs this calculation to check whether the inclusion of balls 2.2 holds.

and

$$B_{\mathfrak{Y}}(s'(t(\lambda)), \tfrac{2}{(\lambda+1)}) \subseteq B. \tag{2.2}$$

If this happens for the word $w\eta^\kappa$ at time step ξ, let

$$\beta = \min\{\gamma \mid \forall \alpha \in \xi \ (H_{\mathrm{in}}(\alpha) \neq \gamma)\},$$

(in words, β is the least index of a cell which the machine for F has not read before time ξ, which is the moment when the machine determines that condition (2.2) holds, so we can be sure that the machine for F has only been able to read its input up to cell β). By Lemma 2.1.4 it follows that if $t \subseteq q$ for some $q \in \mathrm{dom}(\delta_{\mathfrak{Y}})$, then

$$\delta_{\mathfrak{Y}}(q) \in \mathrm{cl}(B_{\mathfrak{Y}}(s'(t(\lambda)), \tfrac{1}{\lambda+1})) \subseteq \overline{B}_{\mathfrak{Y}}(s'(t(\lambda)), \tfrac{1}{\lambda+1}) \subseteq B_{\mathfrak{Y}}(s'(t(\lambda)), \tfrac{2}{(\lambda+1)}) \subseteq B,$$

hence, in particular, $\delta_{\mathfrak{Y}}(q) \in B$. From this it immediately follows that if $p \in \mathrm{dom}(\delta_{\mathfrak{X}})$ is such that $w\eta^\kappa \restriction \beta \subseteq p$ and $\delta_{\mathfrak{X}}(p) \in \mathrm{dom}(f)$, it will be the case that $f(\delta_{\mathfrak{X}}(p)) \in B$. Therefore we can safely say that any point $x \in X$ named by such p could belong to $f^{-1}[B]$. Now let $\mathrm{compat}_{w\eta^\kappa}$ be the set defined as

$$\mathrm{compat}_{w\eta^\kappa} = \bigcap_{\alpha \in \beta} B_{\mathfrak{X}}(s(w\eta^\kappa(\alpha)), \tfrac{1}{\alpha+1}).$$

By Lemmas 2.1.6 and 2.1.8 the points x which admit a name p extending $w\eta^\kappa \restriction \beta$ are given by a set A such that $\mathrm{compat}_{w\eta^\kappa} \subseteq A \subseteq \mathrm{cl}(\mathrm{compat}_{w\eta^\kappa})$ although we will see that performing this procedure for every word of the form $w\eta^\kappa$ allows us to ignore the points on the border of $\mathrm{compat}_{w\eta^\kappa}$. For now, we stipulate that the machine for G computes a $\mathbf{\Sigma}^0_1$-name for all the sets $\mathrm{compat}_{w\eta^\kappa}$ as above and unionizes these names (these operations are computable by Proposition 2.1.12) to obtain a $\mathbf{\Sigma}^0_1$-name for a set B'.

By the argument above $B' \cap \mathrm{dom}(f) \subseteq f^{-1}[B]$. To show the converse inclusion, let $x \in f^{-1}[B]$, and in the notation of Lemma 2.1.10, let $t \in \kappa^\kappa$ be a branch in npn_x. By definition of npn_x, we have that for every $\beta \in \kappa$,

$$x \in \bigcap_{\alpha \in \beta} B_{\mathfrak{X}}(s(t(\alpha)), \tfrac{1}{\alpha+1}).$$

Hence, we can consider the computation of F on t and, if β is the time step at which we learn that $F(t)$ surely belongs to B (i.e. condition (2.2) holds for t' where t' is the portion of output printed by the machine for F on input t by step β), we know that when running G on the word $w = (t \restriction \beta)^\frown t(\beta)^\kappa$, the set compat_w that we will enumerate in the candidate preimage of B will contain x. This shows that $f^{-1}[B] \subseteq B' \cap \mathrm{dom}(f)$ and thus $f^{-1}[B] = B' \cap \mathrm{dom}(f)$. When the machine for G is given an arbitrary open subset $U \subseteq Y$ as a list of open balls $(B_\alpha)_{\alpha\in\kappa}$, it performs the computation described above in parallel for each of these balls and then computes the union of the open sets $(B'_\alpha)_{\alpha\in\kappa}$ to obtain a $\mathbf{\Sigma}^0_1$-name of a set V with $V \cap \mathrm{dom}(f) = f^{-1}[U]$. This shows that G is a computable realizer for Φ, hence f is $\mathbf{\Sigma}^0_1$-computable.

Conversely assume that there exists an operation $\Phi \colon \mathbf{\Sigma}^0_1(\mathfrak{Y}) \rightrightarrows \mathbf{\Sigma}^0_1(\mathfrak{X})$ such that for all $U \in \mathbf{\Sigma}^0_1(\mathfrak{Y})$ and all $V \in \Phi(U)$, $f^{-1}[U] = V \cap \mathrm{dom}(f)$ and that Φ has a computable realizer G. We define a T2κ-computable function F and then prove that it actually realizes f. The machine M for F works in the following recursive manner: on input $p \in \kappa^\kappa$, if at time step β, the word $t \in \kappa^{<\kappa}$ is the current output, M simulates the machine for G in order to compute in parallel $G(O_\alpha)$ for all $\alpha \in \kappa$, where

$$O_\alpha = B_{\mathfrak{Y}}(s'(\alpha), \tfrac{1}{|t|+1}),$$

and it looks for the first ordinal δ such that $x \in G(O_\delta)$[3] and such that $t^\frown\delta \in \mathrm{extw}_{\mathfrak{Y}}$. The intuition behind this construction is that we are building our sequence $F(p)$ by appending bits to the output tape in a way such that the words $w \in \kappa^{<\kappa}$ which appear on the output tape are always elements of $\mathrm{npn}_{f(x)}$.

[3]For every computable κ-metric space, there's a κ-Turing Machine which terminates in fewer than κ steps on input $\langle p, q\rangle$ if and only if the element x with code p belongs to the open set O with code q.

We claim that for every $p \in \mathrm{dom}(f \circ \delta_{\mathfrak{X}})$, M computes indefinitely and in the long run outputs $q \in \mathrm{dom}(\delta_{\mathfrak{Y}})$ such that $\delta_{\mathfrak{Y}}(q) = f(\delta_{\mathfrak{X}}(p))$. Assume $p \in \mathrm{dom}(f \circ \delta_{\mathfrak{X}})$ is a name for $x \in X$ and at some point in the computation of M on input p, the word $t \in \kappa^{<\kappa}$ has been printed on the output tape. We prove that if $t \in \mathrm{npn}_{f(x)}$, then M always finds an ordinal η to append to the currrent output t and that η is such that $t^\frown\eta \in \mathrm{npn}_{f(x)}$. To see why this is enough, let $h_p :\subseteq \kappa^{<\kappa} \to \kappa^{<\kappa}$ be the function given by

$$t \mapsto t^\frown\eta$$

if and only if the machine for F described above appends the ordinal η to the word t when run on input p.

Our claim is then that h_p has the property that for every $t \in \mathrm{npn}_{f(x)}$, $h_p(t) \in \mathrm{npn}_{f(x)}$, so we can appeal to Lemma 1.1.38 (recall that by Lemma 2.1.10, $\mathrm{npn}_{f(x)}$ is $<\kappa$-closed and it coincides with its extendible part) to obtain that the sequence built recursively as

$$r_0 = \emptyset,\ r_{\alpha+1} = h_p(r_\alpha),\ r_\gamma = \bigcup_{\alpha\in\gamma} r_\alpha \text{ for } \gamma \text{ limit ordinal}$$

determines a branch in $\mathrm{npn}_{f(x)}$, hence a $\delta_{\mathfrak{Y}}$-name of $f(x)$ (notice that such name is precisely the output of the computation of M on p).

To prove our claim, let $t \in \kappa^{<\kappa}$ appear on the output tape during the computation of M on p and assume inductively that $t \in \mathrm{npn}_{f(x)}$. The machine M looks for an ordinal η such that $x \in G(O_\eta)$ and $s'(\eta) \in \mathrm{compat}_t$ (both of these requirements are semidecidable, hence M can compute in parallel on all ordinals in κ until it finds some ordinal satisfying both of these). Notice that such an η always exists because $t \in \mathrm{npn}_{f(x)}$ is extendible by Lemma 2.1.10, and the requirements given correspond to asking for an ordinal η such that $t^\frown\eta \in \mathrm{npn}_{f(x)}$. This proves both of our claims at once: first, that M will always find ordinals to append to its output tape and second, that it does so in such a way that the words appearing on the output tape always belong to $\mathrm{npn}_{f(x)}$. By the argument above this implies that M computes indefinitely on input $p \in \mathrm{dom}(f \circ \delta_{\mathfrak{X}})$, producing in the long run a sequence $q \in \kappa^\kappa$ such that $\delta_{\mathfrak{Y}}(q) = f(\delta_{\mathfrak{X}}(p))$. In other words M computes a realizer of f, which is what was to show. ☐

Note that the proof above is uniform in the codes for the computable realizers involved, hence we can refine the result to the following.

Corollary 2.2.3. *Assume Hypothesis 1.2.19 and let $\mathfrak{X} = (X, d, s)$ and $\mathfrak{Y} = (Y, d', s')$ be computable κ-metric spaces and let $f :\subseteq \mathfrak{X} \to \mathfrak{Y}$ be a computable function. There is a κ-Turing machine M which, on input a code α for a computable realizer of f, outputs a code $M(\alpha)$ for a computable realizer of $\Phi\colon \mathbf{\Sigma}^0_1(\mathfrak{Y}) \to \mathbf{\Sigma}^0_1(\mathfrak{X})$. Similarly there is a κ-Turing machine M' which takes codes for computable realizers of Φ and outputs codes for computable realizers of f.*

Another similarity with ordinary $\mathbf{\Sigma}^0_n$-measurable maps is given by the following lemma.

Lemma 2.2.4. *Let $f :\subseteq \mathfrak{X} \rightrightarrows \mathfrak{Y}$ be a $\mathbf{\Sigma}^0_n$-computable operation, then for every $k \in \omega$ there are computable realizers for the operations $\Phi_k\colon \mathbf{\Sigma}^0_{k+1}(\mathfrak{Y}) \rightrightarrows \mathbf{\Sigma}^0_{n+k-1}(\mathfrak{Y})$ such that for all $A \in \mathbf{\Sigma}^0_k(\mathfrak{Y})$ and for all $V \in \Phi_k(A)$*

$$f^{-1}[A] = \mathrm{dom}(f) \cap V.$$

Proof. We prove this by induction on k: the statement for $k = 1$ is the definition of $\mathbf{\Sigma}^0_n$-computability. Now assume that the claim holds for a natural number k and let p be a $\mathbf{\Sigma}^0_{k+2}$-name for $U \in Y$, then each p_α is a $\mathbf{\Pi}^0_{k+1}$-name for a set C_α and $U = \bigcup_{\alpha\in\kappa} C_\alpha$. In turn this means that p_α is a $\mathbf{\Sigma}^0_{k+1}$-name for $A_\alpha = Y \setminus C_\alpha$. By the inductive assumption we can compute a sequence q_α which is a $\mathbf{\Sigma}^0_{n+k-1}$-name of a set V_α such that $f^{-1}[A_\alpha] = \mathrm{dom}(f) \cap V_\alpha$. As this is possible for every α, we compute the sequence t given by $\langle (q_\alpha)_{\alpha\in\kappa}\rangle$. Exploiting the complementation map we can obtain a sequence containing $\mathbf{\Pi}^0_{n+k-1}$-names of V_α^{c}, which is already a $\mathbf{\Sigma}^0_{n+k}$-name for $V' = \bigcup_{\alpha\in\kappa} V_\alpha^{\mathrm{c}}$. Note that

$$f^{-1}[U] = \bigcup_{\alpha\in\kappa} f^{-1}[C_\alpha] = \bigcup_{\alpha\in\kappa} (f^{-1}[A_\alpha])^{\mathrm{c}},$$

and $f^{-1}[A_\alpha] = V_\alpha \cap \mathrm{dom}(f)$, so $f^{-1}[C_\alpha] = V_\alpha^{\mathrm{c}} \cap \mathrm{dom}(f)$ and $f^{-1}[U] = V' \cap \mathrm{dom}(f)$. This shows that Φ_{k+1} is computable, concluding the induction and proving the claim. □

The next two results state that $\mathbf{\Sigma}^0_n$-computable functions are closed under composition and products with computable functions. These are stepping stones for the proof that $\mathbf{\Sigma}^0_n$-computable functions on generalized Baire space are closed under Weihrauch reductions. Note how the corresponding results where the word "computable" is swapped for "measurable" hold trivially.

Lemma 2.2.5. *Let $f :\subseteq \mathfrak{X} \rightrightarrows \mathfrak{Y}$ and $g :\subseteq \mathfrak{Y} \rightrightarrows \mathfrak{Z}$ be operations which are respectively $\mathbf{\Sigma}^0_n$ and $\mathbf{\Sigma}^0_k$-computable and composable. The function $g \circ f :\subseteq \mathfrak{X} \rightrightarrows \mathfrak{Z}$ is $\mathbf{\Sigma}^0_{n+k-1}$-computable.*

Proof. For $n = 1$ the result is trivial. Assume $n > 1$ and let Φ_f be the multivalued function which witnesses the $\mathbf{\Sigma}^0_n$-computability of f, in particular let H be a computable realizer of it. Similarly let $\Phi_{n-2,g} \colon \mathbf{\Sigma}^0_{n-1}(\mathfrak{Z}) \rightrightarrows \mathbf{\Sigma}^0_{n+k-2}(\mathfrak{Y})$ as in Lemma 2.2.4 and let H' be one of its computable realizers. Let p be a code for $U \in \mathbf{\Sigma}^0_1(\mathfrak{Z})$ and let V be the set named by $H(p)$. As $V \in \mathbf{\Sigma}^0_n$ there are sets $(A_\alpha)_{\alpha\in\kappa}$ in $\mathbf{\Pi}^0_{n-1}$ such that $\bigcup_{\alpha\in\kappa} A_\alpha = V$ and $h(U)$ is the sequence of their $\mathbf{\Pi}^0_{n+1}$ names. We obtain

$$(g \circ f)^{-1}[U] = f^{-1}[g^{-1}[U]] = \bigcup_{\alpha\in\kappa} f^{-1}[A_\alpha],$$

and since $f^{-1}[A_\alpha] = (f^{-1}[A_\alpha^{\mathrm{c}}])^{\mathrm{c}}$ and $A_\alpha^{\mathrm{c}} \in \mathbf{\Sigma}^0_{n-1}$, $(f^{-1}[A_\alpha^{\mathrm{c}}])$ is $\mathbf{\Sigma}^0_{n+k-2}$. Hence each $f^{-1}[A_\alpha]$ is $\mathbf{\Pi}^0_{n+k-2}$. This means that we can compute a sequence of $\mathbf{\Pi}^0_{n+k-2}$-names for $f^{-1}[A_\alpha]$ and the union of these sets is $(g\circ f)^{-1}[U]$, so we can compute a $\mathbf{\Sigma}^0_{n+k-1}$ name for a set W such that $W \cap \mathrm{dom}(g\circ f) = (g\circ f)^{-1}[U]$. Since this can be done uniformly in (the code for) U, this shows that $g \circ f$ is $\mathbf{\Sigma}^0_{n+k-1}$-computable. □

Lemma 2.2.6. *Let $f :\subseteq \mathfrak{X} \rightrightarrows \mathfrak{Y}$ and $g :\subseteq \mathfrak{U} \rightrightarrows \mathfrak{V}$ be $\mathbf{\Sigma}^0_n$-computable operations, then $f \times g :\subseteq \mathfrak{X} \times \mathfrak{U} \rightrightarrows \mathfrak{Y} \times \mathfrak{V}$ is $\mathbf{\Sigma}^0_n$-computable.*

Proof. For $A \in \mathbf{\Sigma}^0_1(\mathfrak{Y})$ and $B \in \mathbf{\Sigma}^0_1(\mathfrak{V})$, we have that $(f \times g)^{-1}[A \times B] = f^{-1}[A] \times g^{-1}[B]$. By $\mathbf{\Sigma}^0_n$-computability of each operation, we can compute names for sets C, D such that $f^{-1}[A] = \mathrm{dom}(f) \cap C$ and $g^{-1}[B] = \mathrm{dom}(g) \cap D$, so a name for a set E such that $E \cap \mathrm{dom}(f \times g) = (f \times g)^{-1}[A \times B]$ is obtained by considering the product $C \times D$, which is possible by Proposition 2.1.12. □

2.3 A family of functions complete for $\mathbf{\Sigma}^0_n$-measurable maps

We introduce a family of maps indexed on the natural numbers $C_n \colon \kappa^\kappa \to \kappa^\kappa$ which are an adaptation of the maps found in [5] with the same name and we prove that each C_n is complete with respect to $\leq_{\mathrm{W}}$ for the set of partial $\mathbf{\Sigma}^0_{n+1}$-computable single valued maps on κ^κ. At the same time we show that each C_n is complete with respect to $\leq_{\mathrm{tW}}$ for the set of partial $\mathbf{\Sigma}^0_{n+1}$-measurable single valued maps on κ^κ. We close this section with an explanation of the relation between our maps and the family $(C_{U^\alpha})_{\alpha\in\kappa}$ of [16], which is another generalization of the family in [5].

2.3.1 Reductions for computable κ-metric spaces

We introduce a notion of reduction for functions on computable κ-metric spaces which closely resembles Weihrauch reduction. This type of reductions is introduced for technical convenience, since, as we will see, it induces a reducibility relation which turns out to coincide with Weihrauch reducibility, both in the classical and in the generalized context (see see [5, Lemma 7.4] for the proof of equivalence in the classical context, and Corollary 2.3.6 for the proof in the generalized context).

Definition 2.3.1 (Alternative notion of reductions)**.** *Let $f :\subseteq \mathfrak{X} \to \mathfrak{Y}$ and $g :\subseteq \mathfrak{U} \to \mathfrak{V}$ be functions between computable κ-metric spaces, we say that*

- *$f \leq_{\mathrm{t}} g$ if there exist continuous functions $A :\subseteq \mathfrak{X} \times \mathfrak{V} \to \mathfrak{Y}$ and $B :\subseteq \mathfrak{X} \to \mathfrak{U}$ such that for all $x \in \mathrm{dom}(f)$:*
$$f(x) = A(x, g(B(x)))$$

- $f \leq_{\mathrm{c}} g$ *if the functions* A, B *above are additionally computable with respect to the respective Cauchy representations.*

Following [5], we will prove our Completeness Theorem using the reducibilities $\leq_{\mathrm{t}}$ and $\leq_{\mathrm{c}}$ (cf. Theorem 5.5 in the paper mentioned). In light of the equivalence mentioned above, our Completeness Theorem also holds in terms of $\leq_{\mathrm{W}}$ and $\leq_{\mathrm{tW}}$. We mention here that in [5], the author actually proves a second Completeness Theorem [5, Theorem 7.6] which shows that each C_n is Weihrauch complete with respect to the class of total single valued $\mathbf{\Sigma}^0_n$-measurable functions between any computable metric spaces. This is based on the first Completeness Theorem and the so-called Representation Theorem [5, Theorem 6.1], which states that total single valued functions on computable metric spaces are $\mathbf{\Sigma}^0_n$-(effectively) measurable if and only if they admit a $\mathbf{\Sigma}^0_n$-(effectively) measurable realizer.[4] We point out that a proof of the Representation Theorem in the generalized context would immediately lead to a strengthening of our Completeness Theorem.

We prove a result on the Cauchy representation of κ^κ which will be useful in establishing the relation between these alternative reducibilities and (topological) Weihrauch reducibility.

Lemma 2.3.2. *Let* $\delta_{\kappa^\kappa} :\subseteq \kappa^\kappa \to \kappa^\kappa$ *be the Cauchy representation of* κ^κ*, considered as a computable* κ*-metric space as above. Under the assumption of Hypothesis 1.2.19, the function* δ_{κ^κ} *is* $[\delta_{\kappa^\kappa}, \delta_{\kappa^\kappa}]$*-computable. Moreover, it has a* $[\delta_{\kappa^\kappa}, \delta_{\kappa^\kappa}]$*-computable right inverse* $\delta^{-1}_{\kappa^\kappa}$*. The same holds if we replace* κ^κ *with* 2^κ*.*

Proof. We only prove the lemma for κ^κ, the case of 2^κ being analogous.
It is clear that δ_{κ^κ} itself is a realizer of δ_{κ^κ}, hence it suffices to prove that δ_{κ^κ} is a computable function with respect to the ordinary notion of computability available on κ^κ. To that end, notice that by definition of fast convergent Cauchy sequences (and by definition of the metric on κ^κ), $\delta_{\kappa^\kappa}(p) = x$ if and only if for all $\alpha \in \kappa$, we have that $\mu(p(\alpha))\restriction(\alpha+1) \subseteq x$. Now consider the machine M which works as follows: given p as input, when reading the ordinal $p(\alpha)$, M checks whether for all $\beta < \alpha$

$$\mu(p(\beta))\restriction(\beta+1) \subseteq \mu(p(\alpha))\restriction\alpha.$$

If this is not the case, it halts. If this is actually the case, it prints bits on the output tape until the latter contains $\mu(p(\alpha))\restriction\alpha$. This machine computes indefinitely on all sequences $p \in \mathrm{dom}(\delta_{\kappa^\kappa})$ (and only on those) and outputs $\delta_{\kappa^\kappa}(p)$, so it computes the Cauchy representation δ_{κ^κ}. To show that δ_{κ^κ} admits a computable right inverse, consider the κ-Turing machine M that works as follows: given input p, print the unique $q_p \in \kappa^\kappa$ such that $\mu(q(\alpha)) = p\restriction(\alpha+1)$. This is a computable procedure because μ admits a computable right inverse. For every $p \in \kappa^\kappa$, the sequence q_p is a δ_{κ^κ}-name of p, so M computes a realizer for $\delta^{-1}_{\kappa^\kappa}$. □

This can be rephrased as:

Corollary 2.3.3. *Assuming Hypothesis 1.2.19, the representations for generalized Baire space* δ_{κ^κ} *and* $\mathrm{id}_{\kappa^\kappa}$ *are computably equivalent. The same holds for* δ_{2^κ} *and* id_{2^κ}*.*

Proof. Again we only prove the claim for κ^κ as the case of 2^κ is similar. We need to provide two Type 2 computable functions f and g such that for every $p \in \mathrm{dom}(\delta_{\kappa^\kappa})$, we have $p \in \mathrm{dom}(f)$ and $f(p) = \mathrm{id}_{\kappa^\kappa}(f(p)) = \delta_{\kappa^\kappa}(p)$; and for every $q \in \mathrm{dom}(\mathrm{id}_{\kappa^\kappa})$, we have $q \in \mathrm{dom}(g)$ and $\delta_{\kappa^\kappa}(g(q)) = \mathrm{id}_{\kappa^\kappa}(q) = q$. Let $f = \delta_{\kappa^\kappa}$ and $g = \delta^{-1}_{\kappa^\kappa}$: these are computable functions by the proof of Lemma 2.3.2 and they clearly fulfill the requirements. □

This immediately yields:

Corollary 2.3.4. *Let* $f :\subseteq \kappa^\kappa \to \kappa^\kappa$*, then, under the assumption of Hypothesis 1.2.19,* f *is* $[\delta_{\kappa^\kappa}, \delta_{\kappa^\kappa}]$*-computable if and only if it is a Type 2 computable function.*

[4]Note that this result encompasses the Main Theorem of Computable Analysis for computable metric spaces as well as [4, Theorem 6.2] (the classical analogue of our Theorem 2.2.2), and it is a strong piece of evidence for the naturalness of the concept of $\mathbf{\Sigma}^0_n$-(effective) measurability in the context of computable analysis.

Proof. Follows from Corollary 2.3.3 and the fact that the notions of Type 2 computability and $[\mathrm{id}_{\kappa^\kappa}, \mathrm{id}_{\kappa^\kappa}]$-computability obviously coincide. $\square$

We can now prove the following result, which appears in the ω-context in [5] as Lemma 7.3.

Lemma 2.3.5. *Let $f :\subseteq \mathfrak{X} \to \mathfrak{Y}$ and $g :\subseteq \mathfrak{U} \to \mathfrak{V}$ be functions between computable κ-metric spaces, then*

$$f \circ \delta_{\mathfrak{X}} \leq_{\mathrm{t}} g \circ \delta_{\mathfrak{U}} \iff f \leq_{\mathrm{tW}} g.$$

An analogous statement with $\leq_{\mathrm{t}}$ replaced by $\leq_{\mathrm{c}}$ and $\leq_{\mathrm{tW}}$ replaced with $\leq_{\mathrm{W}}$ holds, under the assumption of Hypothesis 1.2.19.

Proof. For the computable case, assume Hypothesis 1.2.19 and assume $f \circ \delta_{\mathfrak{X}} \leq_{\mathrm{c}} g \circ \delta_{\mathfrak{U}}$, then there are computable functions (with respect to the respective Cauchy representations) $A' :\subseteq \kappa^\kappa \times \mathfrak{V} \to \mathfrak{Y}$ and $B :\subseteq \kappa^\kappa \to \kappa^\kappa$ such that for all $p \in \mathrm{dom}(f \circ \delta_{\mathfrak{X}})$, $p \in \mathrm{dom}(g \circ \delta_{\mathfrak{U}} \circ B)$ and $f \circ \delta_{\mathfrak{X}}(p) = A'(p, g \circ \delta_{\mathfrak{U}} \circ B(p))$. Since A' is computable, it has a computable realizer $A :\subseteq \kappa^\kappa \times \kappa^\kappa \to \kappa^\kappa$ (so A is a Type 2 computable function), such that for all $(p, q) \in \mathrm{dom}(A' \circ (\delta_{\kappa^\kappa} \times \delta_{\mathfrak{V}}))$ we have that $\delta_{\mathfrak{Y}}(A(p, q)) = A'(\delta_{\kappa^\kappa}(p), \delta_{\mathfrak{V}}(q))$. Now let G be a realizer of g, so $\delta_{\mathfrak{V}}(G(t)) = g(\delta_{\mathfrak{U}}(t))$ for all $t \in \mathrm{dom}(g \circ \delta_{\mathfrak{U}})$. We obtain, for all $p \in \mathrm{dom}(f \circ \delta_{\mathfrak{X}})$:

$$f(\delta_{\mathfrak{X}}(p)) = A'(p, g \circ \delta_{\mathfrak{U}} \circ B(p)) = A'(p, \delta_{\mathfrak{V}} \circ G \circ B(p)) = \delta_{\mathfrak{Y}}(A(\delta_{\kappa^\kappa}^{-1}(p), G(B(p)))).$$

Hence the partial function $F :\subseteq \kappa^\kappa \to \kappa^\kappa$ with $\mathrm{dom}(F) = \mathrm{dom}(f \circ \delta_{\mathfrak{X}})$ and $F(p) = A(\delta_{\kappa^\kappa}^{-1}(p), GB(p))$ is a realizer of f. Note that by Lemma 2.3.4, B is actually a Type 2 computable function, hence $A \circ (\delta_{\kappa^\kappa}^{-1} \times \mathrm{id}_{\kappa^\kappa})$ and B witness a Weihrauch reduction $f \leq_{\mathrm{W}} g$.

Conversely assume that $f \leq_{\mathrm{W}} g$, so let $H :\subseteq \kappa^\kappa \times \kappa^\kappa \to \kappa^\kappa$ and $K :\subseteq \kappa^\kappa \to \kappa^\kappa$ be Type 2 computable functions such that for all realizers G of g and all $p \in \mathrm{dom}(f \circ \delta_{\mathfrak{X}})$, the function

$$p \mapsto H(p, G \circ K(p))$$

realizes f, i.e. $f(\delta_{\mathfrak{X}}(p)) = \delta_{\mathfrak{Y}}(H(p, G \circ K(p)))$. Notice that again by Lemma 2.3.4, K is also computable with respect to the Cauchy representation on κ^κ. Let $R\colon \mathfrak{V} \to \kappa^\kappa$ be a right inverse of $\delta_{\mathfrak{V}}$ and let $H'\colon \kappa^\kappa \times \mathfrak{V} \to \mathfrak{Y}$ be defined as $H'(p, v) = \delta_{\mathfrak{Y}}(H(p, R(v)))$ for all (p, v) such that $p \in \mathrm{dom}(f \circ \delta_{\mathfrak{X}})$ and $v = g \circ \delta_{\mathfrak{U}} \circ K(p)$. We obtain

$$H'(p, \delta_{\mathfrak{V}}(q)) = \delta_{\mathfrak{Y}}(H(p, R \circ \delta_V(q)))$$

for all $p \in \mathrm{dom}(f \circ \delta_{\mathfrak{X}})$ and all $q \in \delta_{\mathfrak{V}}^{-1}[\{v\}]$ with $v = g \circ \delta_{\mathfrak{U}} \circ K(p)$. Notice that $R \circ \delta_{\mathfrak{V}}(q)$ is some sequence t such that $\delta_{\mathfrak{V}}(t) = v$. If we fix p and let $v = g \circ \delta_{\mathfrak{U}} \circ K(p)$ and we pick a realizer G of g such that $t = G(K(p))$, we obtain

$$f(\delta_{\mathfrak{X}}(p)) = \delta_{\mathfrak{Y}}(H(p, t)) = \delta_{\mathfrak{Y}}(H(p, R \circ \delta_{\mathfrak{V}}(q))) = H'(p, \delta_{\mathfrak{V}}(q))$$

for all $q \in \delta_{\mathfrak{V}}^{-1}[\{v\}]$. Now let G' be another realizer of g such that $q = G(K(p))$, then

$$\delta_{\mathfrak{Y}}(H(p, t)) = \delta_{\mathfrak{Y}}(H(p, q))$$

because both expression must be equal to $f(\delta_{\mathfrak{X}}(p))$. Hence $H'(p, \delta_{\mathfrak{V}}(q)) = \delta_{\mathfrak{Y}}(H(p, q))$ for all $p \in \mathrm{dom}(f \circ \delta_{\mathfrak{X}})$ and all $q \in \delta_{\mathfrak{V}}^{-1}[\{v\}]$ where $v = g \circ \delta_{\mathfrak{U}} \circ K(p)$ and consequently, if $p' \in \delta_{\kappa^\kappa}^{-1}[\{p\}]$ we obtain

$$H'(\delta_{\kappa^\kappa}(p'), \delta_{\mathfrak{V}}(q)) = \delta_{\mathfrak{Y}}(H(p, q)) = \delta_{\mathfrak{Y}}(H(\delta_{\kappa^\kappa}(p'), q)).$$

In other words $H \circ (\delta_{\kappa^\kappa} \times \mathrm{id}_{\kappa^\kappa})$ is a computable realizer of H'. Now fix a realizer G of g and define $G' = R \circ \delta_{\mathfrak{V}} \circ G$, then $\delta_{\mathfrak{V}} \circ G' = \delta_{\mathfrak{V}} \circ G = g \circ \delta_{\mathfrak{U}}$, therefore G' realizes g, moreover, we have

$$f \circ \delta_{\mathfrak{X}}(p) = \delta_{\mathfrak{Y}}(H(p, G'(K(p)))) = \delta_{\mathfrak{Y}}(H(p, R \circ \delta_{\mathfrak{V}} \circ G \circ K(p))) = H'(p, \delta_{\mathfrak{V}} \circ G \circ K(p))$$

for all $p \in \mathrm{dom}(f \circ \delta_{\mathfrak{X}})$, hence $f \circ \delta_{\mathfrak{X}} \leq \delta_{\mathfrak{V}} \circ G = g \circ \delta_{\mathfrak{U}}$, proving the claim. The continuous case is proved analogously. $\square$

Corollary 2.3.6. *Let $f, g :\subseteq \kappa^\kappa \to \kappa^\kappa$ be partial functions, then,*

$$f \leq_{\mathrm{t}} g \iff f \leq_{\mathrm{tW}} g.$$

An analogous statement with $\leq_{\mathrm{t}}$ replaced by $\leq_{\mathrm{c}}$ and $\leq_{\mathrm{tW}}$ replaced with $\leq_{\mathrm{W}}$ holds under the assumption of Hypothesis 1.2.19.

Proof. By Lemma 2.3.5 we just need to show that $f \leq_{\mathrm{c}} g \iff f \circ \delta_{\kappa^\kappa} \leq_{\mathrm{c}} g \circ \delta_{\kappa^\kappa}$. Assume that $f \leq_{\mathrm{c}} g$, so assume there are A, B computable such that $f(x) = A(x, g(B(x)))$ for all $x \in \mathrm{dom}(f)$. We obtain that, for all $p \in \mathrm{dom}(f \circ \delta_{\kappa^\kappa})$,

$$f(\delta_{\kappa^\kappa}(p)) = A(\delta_{\kappa^\kappa}(p), g(B(\delta_{\kappa^\kappa}(p)))) = A(\delta_{\kappa^\kappa}(p), g \circ \delta_{\kappa^\kappa} \circ \delta_{\kappa^\kappa}^{-1}(B(\delta_{\kappa^\kappa}(p))).$$

We can then define the computable functions $A'(x, y) = A(\delta_{\kappa^\kappa}(x), y)$ and $B'(x) = \delta_{\kappa^\kappa}^{-1} \circ B \circ \delta_{\kappa^\kappa}(x)$ and obtain that $f \circ \delta_{\kappa^\kappa} \leq_{\mathrm{c}} g \circ \delta_{\kappa^\kappa}$.

Conversely, if for all $p \in \mathrm{dom}(f \circ \delta_{\kappa^\kappa})$ we have

$$f \circ \delta_{\kappa^\kappa}(p) = A(p, g \circ \delta_{\kappa^\kappa}(B(p))),$$

for computable functions A, B, then if $x \in \mathrm{dom}(f)$ and $p' = \delta_{\kappa^\kappa}^{-1}(x)$, we obtain

$$f(x) = A(p', g \circ \delta_{\kappa^\kappa}(B(p'))) = A(\delta_{\kappa^\kappa}^{-1}(x), g \circ \delta_{\kappa^\kappa}(B(\delta_{\kappa^\kappa}^{-1}(x)))).$$

We can then define $A'(x, y) = A(\delta_{\kappa^\kappa}^{-1}(x), y)$ and $B'(x) = B \circ \delta_{\kappa^\kappa}^{-1}(x)$ and we obtain that $f \leq_{\mathrm{c}} g$. ☐

Putting everything together we obtain that, for single-valued partial functions on generalized Baire space, $\leq_{\mathrm{c}}$ coincides with $\leq_{\mathrm{W}}$ and $\leq_{\mathrm{t}}$ coincides with $\leq_{\mathrm{tW}}$. This can be in turn used to prove the closure of the $\mathbf{\Sigma}^0_n$-measurable partial functions on generalized Baire space under Weihrauch reducibility.

Proposition 2.3.7. *Let $f :\subseteq \kappa^\kappa \to \kappa^\kappa$ and $g :\subseteq \kappa^\kappa \to \kappa^\kappa$ and assume $f \leq_{\mathrm{W}} g$ (we assume κ^κ is represented as a computable κ-metric space). For all $n \in \omega$, we obtain that assuming Hypothesis 1.2.19, if g is $\mathbf{\Sigma}^0_n$-computable, then so is f.*

Proof. By Corollary 2.3.6, $f \leq_{\mathrm{W}} g$ means that there are computable functions A, B such that $f(x) = A(x, g(B(x)))$ for all $x \in \mathrm{dom}(f)$. Lemmas 2.2.6 and 2.2.5, together with Theorem 2.2.2 immediately yield that f must also be $\mathbf{\Sigma}^0_n$-computable. ☐

This proposition justifies working with the easier $\leq_{\mathrm{c}}/\leq_{\mathrm{t}}$ reductions instead of Weihrauch reductions when talking about functions on generalized Baire space. Note that the constructions of Lemma 2.3.5, Corollary 2.3.6 and Proposition 2.3.7 relativize to any oracle, therefore we have results analogous to these mentioned where our notion of computability is relative to any $o \in \kappa^\kappa$.

2.3.2 Effective logical normal form for subsets of generalized Baire space

One last building block which will be needed in the proof of the main result of the section is the following: we can computably describe sets in $\mathbf{\Sigma}^0_n(\kappa^\kappa)$ in so-called Logical Normal Form. This is a rather technical result whose proof is largely based on symbolic manipulations. In this section we spell out the details of the proof, which closely follows the classical proof (see [5, Lemma 5.4]).

We split the proof for readability, so we first give two intermediate results as lemmas.

Lemma 2.3.8. *Let $n \geq 1$ and for every sequence $(A_\alpha)_{\alpha\in\kappa} \subseteq \mathbf{\Sigma}^0_n(\kappa^\kappa)$, define*

$$A = \bigcup_{\alpha\in\kappa} (A_\alpha \times \{\alpha\}) \in \mathbf{\Sigma}^0_n(\kappa^\kappa \times \kappa).$$

The corresponding operation

$$(\mathbf{\Sigma}^0_n(\kappa^\kappa))^\kappa \to \mathbf{\Sigma}^0_n(\kappa^\kappa \times \kappa),\ (A_\alpha)_{\alpha\in\kappa} \mapsto A$$

is computable relative to the respective representations.

Similarly, if $(B_\alpha)_{\alpha\in\kappa}$ *is a sequence of* $\mathbf{\Pi}^0_n(\kappa^\kappa)$ *sets, define*

$$B = \bigcup_{\alpha\in\kappa}(B_\alpha \times \{\alpha\}) \in \mathbf{\Pi}^0_n(\kappa^\kappa \times \kappa).$$

The corresponding operation

$$(\mathbf{\Pi}^0_n(\kappa^\kappa))^\kappa \to \mathbf{\Pi}^0_n(\kappa^\kappa \times \kappa),\ (B_\alpha)_{\alpha\in\kappa} \mapsto B$$

is computable relative to the respective representations.

Proof. We prove both claims by a simultaneous induction on n: if $n = 1$, notice that for every α, the operation $\mathbf{\Sigma}^0_1(\kappa^\kappa) \to \mathbf{\Sigma}^0_1(\kappa^\kappa \times \kappa)$ which maps A to $A \times \{\alpha\}$ is trivially computable (uniformly in α), hence the operation h which maps $(A_\alpha)_{\alpha\in\kappa}$ to $(A_\alpha \times \{\alpha\})_{\alpha\in\kappa}$ is a computable. This implies that the operation $(A_\alpha \times \{\alpha\})_{\alpha\in\kappa} \mapsto \bigcup_{\alpha\in\kappa}(A_\alpha \times \{\alpha\})$ is computable as it is the composition of h and the κ-union operation of Proposition 2.1.12.

For the statement with $\mathbf{\Pi}^0_1(\kappa^\kappa)$ sets, it suffices to notice that

$$\left(\bigcup_{\alpha\in\kappa}(A_\alpha \times \{\alpha\})\right)^{\mathrm{c}} = \bigcup_{\alpha\in\kappa}(A^{\mathrm{c}}_\alpha \times \{\alpha\}).$$

Since we want a closed name for $\bigcup_{\alpha\in\kappa}(A_\alpha \times \{\alpha\})$, we just need an open name for its complement. The equivalence above tells us that we need an open name for the right hand side. Notice that a $(\mathbf{\Pi}^0_1(\kappa^\kappa))^\kappa$-name of $(A_\alpha)_{\alpha\in\kappa}$ is precisely a $(\mathbf{\Sigma}^0_1(\kappa^\kappa))^\kappa$-name for $(A^{\mathrm{c}}_\alpha)_{\alpha\in\kappa}$ and from it we can computably find a $\mathbf{\Sigma}^0_1(\kappa^\kappa \times \kappa)$ name for $\bigcup_{\alpha\in\kappa}(A^{\mathrm{c}}_\alpha \times \{\alpha\})$, which is just what we needed to obtain.

Now let $n \geq 1$ and by induction assume that both claims hold for level n. Let $(A_\alpha)_{\alpha\in\kappa}$ be a sequence of $\mathbf{\Sigma}^0_{n+1}(\kappa^\kappa)$ sets. For each α, there are sets $(C^\alpha_\gamma)_{\gamma\in\kappa} \in \mathbf{\Pi}^0_n(\kappa^\kappa)$ such that

$$A_\alpha = \bigcup_{\gamma\in\kappa} C^\alpha_\gamma,$$

and the given names for the A_αs are made out of $\mathbf{\Pi}^0_n$-names for the C^α_γs. We have

$$\bigcup_{\alpha\in\kappa}(A_\alpha \times \{\alpha\}) = \bigcup_{\alpha\in\kappa}\bigcup_{\gamma\in\kappa}(C^\alpha_\gamma \times \{\alpha\}) = \bigcup_{\gamma\in\kappa}\bigcup_{\alpha\in\kappa}(C^\alpha_\gamma \times \{\alpha\}).$$

By inductive assumption we can computably and uniformly find $\mathbf{\Pi}^0_n(\kappa^\kappa \times \kappa)$-names for each of $\bigcup_{\alpha\in\kappa} C^\alpha_\gamma \times \{\alpha\}$ and interleaving these we obtain a $\mathbf{\Sigma}^0_{n+1}(\kappa^\kappa \times \kappa)$-name for $\bigcup_{\alpha\in\kappa}(A_\alpha \times \{\alpha\})$.
For the $\mathbf{\Pi}^0_{n+1}$-case let $(A_\alpha)_{\alpha\in\kappa}$ be a sequence of $\mathbf{\Pi}^0_{n+1}(\kappa^\kappa)$ sets, again we have

$$\left(\bigcup_{\alpha\in\kappa} A_\alpha \times \{\alpha\}\right)^{\mathrm{c}} = \bigcup_{\alpha\in\kappa}(A^{\mathrm{c}}_\alpha \times \{\alpha\}),$$

and we can compute names for the A^{c}_αs, hence we can compute a $\mathbf{\Sigma}^0_{n+1}$-name for $\left(\bigcup_{\alpha\in\kappa} A_\alpha \times \{\alpha\}\right)^{\mathrm{c}}$, which is the $\mathbf{\Pi}^0_{n+1}$-name of $\bigcup_{\alpha\in\kappa}(A_\alpha \times \{\alpha\})$ we wanted. $\square$

Lemma 2.3.9. *For any* $n \geq 1$*, the operation* $\mathbf{\Pi}^0_n(\kappa^\kappa) \to \mathbf{\Sigma}^0_{n+1}(\kappa^\kappa)$ *given by*

$$B \mapsto \{p \in \kappa^\kappa \mid \exists\alpha\langle p, \alpha\rangle \in B\}$$

is computable.

Proof. Let $B \in \mathbf{\Pi}^0_n(\kappa^\kappa)$ and let $C = \{(p,\alpha) \mid \langle p,\alpha\rangle \in B\}$, then $C \in \mathbf{\Pi}^0_n(\kappa^\kappa \times \kappa)$ as it is the preimage of B under the computable map $\langle\cdot,\cdot\rangle$, and given a $\mathbf{\Pi}^0_n(\kappa^\kappa)$-name for B, by Theorem 2.2.2 we can compute a $\mathbf{\Pi}^0_n(\kappa^\kappa \times \kappa)$-name for C.

Given $\alpha \in \kappa$, we let $C_\alpha = \pi_1[C \cap \kappa^\kappa \times \{\alpha\}]$. We want to compute a $\mathbf{\Pi}^0_n(\kappa^\kappa)$-name for each C_α. To do so, first we compute a sequence p_α such that $\delta_{\mathbf{\Pi}^0_n(\kappa^\kappa\times\kappa)}(p_\alpha) = C \cap \kappa^\kappa \times \{\alpha\}$. Now notice that the family of computable homeomorphisms $(\varphi_\alpha)_{\alpha\in\kappa} = ((\pi_1\restriction(\kappa^\kappa \times \{\alpha\}))^{-1})_{\alpha\in\kappa}$ is uniformly computably realized by the function $\langle\cdot,\cdot\rangle$, i.e. for all $\alpha \in \kappa$, the function $\langle\cdot,\alpha\rangle\colon \kappa^\kappa \to \kappa^\kappa$ is a computable realizer for φ_α. This implies that, exploiting Corollary 2.2.3 there exists a κ-Turing Machine which, on input α, outputs a code for a computable realizer of the preimage function of φ_α. Hence, for every α, we can uniformly compute a $\mathbf{\Pi}^0_n(\kappa^\kappa)$-name for C_α and obtain a sequence q which is a $(\mathbf{\Pi}^0_n(\kappa^\kappa))^\kappa$-name for $(C_\alpha)_{\alpha\in\kappa}$. We have that

$$\{p \in \kappa^\kappa \mid \exists\alpha\langle p,\alpha\rangle \in B\} = \bigcup_{\alpha\in\kappa} C_\alpha,$$

hence we can apply the κ-union map to $(C_\alpha)_{\alpha\in\kappa}$ to obtain a $\mathbf{\Sigma}^0_{n+1}$-name for $\{p \in \kappa^\kappa \mid \langle p,\alpha\rangle \in B\}$. This is a computable procedure which is uniform in the code for B, hence we obtain that the operation described is indeed computable. □

We can finally prove the computable Logical Normal Form result for κ^κ:

Proposition 2.3.10 (Logical normal form)**.** *For any $n \geq 1$, the maps*

$$L_n\colon \mathbf{\Delta}^0_1(\kappa^\kappa) \to \mathbf{\Sigma}^0_n(\kappa^\kappa)$$

given by

$$A \mapsto \{p \in \kappa^\kappa \mid \exists\alpha_n\forall\alpha_{n-1}\ldots\langle p,\alpha_1,\ldots,\alpha_n\rangle \in A\}$$

and

$$L'_n\colon \mathbf{\Delta}^0_1(\kappa^\kappa) \to \mathbf{\Pi}^0_n(\kappa^\kappa)$$

given by

$$A \mapsto \{p \in \kappa^\kappa \mid \forall\alpha_n\exists\alpha_{n-1}\ldots\langle p,\alpha_1,\ldots,\alpha_n\rangle \in A\}$$

are surjective, computable and they admit computable multi-valued right inverses.

Proof. We prove that L_n and L'_n are computable for all n by a simultaneous induction: for $n = 1$, we first show that the map given by

$$A \mapsto L_1(A) = \{p \in \kappa^\kappa \mid \exists\alpha\langle p,\alpha\rangle \in A\},$$

for all $A \in \mathbf{\Delta}^0_1(\kappa^\kappa)$ is computable. Note that $L_1(A) = \bigcup_{\alpha\in\kappa} A_\alpha$ where

$$A_\alpha = \{p \in \kappa^\kappa \mid \langle p,\alpha\rangle \in A\},$$

i.e. $A_\alpha = \langle\cdot,\alpha\rangle^{-1}[A]$. The maps $(\langle\cdot,\alpha\rangle\colon \kappa^\kappa \to \kappa^\kappa)_{\alpha\in\kappa}$ are uniformly computable, bijective and they have a computable inverse. Hence, they are uniformly $\mathbf{\Sigma}^0_1$-computable, so $A_\alpha \in \mathbf{\Delta}^0_1$ for all α and we can compute a name for the sequence $(A_\alpha)_{\alpha\in\kappa}$ from the $\mathbf{\Delta}^0_1$-name for A. This implies that $L_1(A)$ is open and, using the κ-union map, we can compute an open name for it, i.e., L_1 is computable. The map $L'_1\colon \mathbf{\Delta}^0_1(\kappa^\kappa) \to \mathbf{\Pi}^0_1(\kappa^\kappa)$ given by $A \mapsto \{p \in \kappa^\kappa \mid \forall\alpha\langle p,\alpha\rangle \in A\}$ is computable as $L'_1(A) = (L_1(A^{\mathrm{c}}))^{\mathrm{c}}$ for all $A \in \mathbf{\Delta}^0_1(\kappa^\kappa)$.

Now let $n > 1$ and assume by induction that the maps L_n and L'_n are computable, and let $A \in \mathbf{\Delta}^0_1(\kappa^\kappa)$. To see that we can compute a name for

$$L_{n+1}(A) = \{p \in \kappa^\kappa \mid \exists\alpha_{n+1}\forall\alpha_n\ldots\langle p,\alpha_1,\ldots,\alpha_{n+1}\rangle \in A\},$$

it suffices to use observe that $L_n = h \circ L'_n$ where h is the function in Lemma 2.3.9. Again we obtain that L'_{n+1} is computable as $L'_{n+1}(A) = (L_{n+1}(A^c))^{\mathrm{c}}$ for all $A \in \mathbf{\Delta}^0_1(\kappa^\kappa)$.

What is left to show is that each of the L_ns and L'_ns is surjective and it has a computable right inverse. Clearly it suffices to show the existence of the computable right inverse. We show the existence of computable realizers of each right inverse, and again we tackle the problem by induction on n: for $n=1$, let $A \in \mathbf{\Sigma}^0_1(\kappa^\kappa)$. Recall that a name for A is given to us as a sequence p such that

$$A = \bigcup_{\ulcorner \alpha,\beta \urcorner \in \mathrm{ran}(p)} B(\mu'(\alpha), \overline{\beta}).$$

Note that in generalized Baire space, basic open balls are also closed, moreover, if the radius $r \notin \{1/(\alpha+1) \mid \alpha \in \kappa\}$, the open ball $B(x,r)$ coincides with the closed ball $\overline{B}(x,r)$. If $r = 1/(\alpha+1)$ for some $\alpha \in \kappa$, then the open ball $B(x,r)$ coincides with the closed ball $\overline{B}(x,r')$ where $r' \in \mathbb{Q}_\kappa$ is a rational such that $r > r' > 1/(\alpha+2)$. Such r' can be explicitly computed as $\overline{\beta}$ where β is the least ordinal coding a rational strictly between r and $1/(\alpha+2)$. Therefore we can (uniformly) compute an alternative $\mathbf{\Sigma}^0_1(\kappa^\kappa)$-name p' of A such that p' codes balls $B(\mu'(\alpha), \overline{\beta})$ with $\overline{\beta} \notin \{1/(\alpha+1) \mid \alpha \in \kappa\}$. Now let $B = \bigcup_{\ulcorner \alpha,\beta \urcorner \in \mathrm{ran}(p')} B(\mu'(\alpha), \overline{\beta}) \times \{\ulcorner \alpha, \beta \urcorner\}$. By Lemma 2.3.8 we can compute a name $\mathbf{\Sigma}^0_1$-name for B. By the proof of point (d) in Proposition 2.1.12 we can then uniformly compute $\mathbf{\Pi}^0_1$-names for each of the balls $B(\mu'(\alpha), \overline{\beta}') = \overline{B}(\mu'(\alpha), \overline{\beta}')$. Using these names and again applying Lemma 2.3.8 we obtain a $\mathbf{\Pi}^0_1(\kappa^\kappa \times \kappa)$-name for B. Putting everything together, we compute a $\mathbf{\Delta}^0_1(\kappa^\kappa \times \kappa)$-name for B. We can then define

$$U = \langle B \rangle = \{p \in \kappa^\kappa \mid p = \langle p', \alpha \rangle \text{ with } (p', \alpha) \in B\},$$

and by computability of $\langle \cdot, \cdot \rangle^{-1}$, we can obtain a name $\mathbf{\Delta}^0_1$-name for U from one for B. We have that $L_1(U) = A$ and a name for U can be uniformly computed from one for A, which is what we wanted to show.

To show that the same holds for L'_1, let $B \in \mathbf{\Pi}^0_1(\kappa^\kappa)$ and let p be a $\mathbf{\Pi}^0_1$-name for it. By definition p is also a $\mathbf{\Sigma}^0_1$-name for B^{c}, hence we can use it to compute a $\mathbf{\Delta}^0_1$-name for a set U such that $L_1(U) = B^{\mathrm{c}}$. Again we exploit the relation between L_1 and L'_1: we have that $(L_1(A))^{\mathrm{c}} = L'_1(A^{\mathrm{c}})$, hence $B = (L_1(U))^{\mathrm{c}} = L'_1(U^{\mathrm{c}})$. Since we can compute a $\mathbf{\Delta}^0_1$-name for U^{c} from a name for U and we can compute a $\mathbf{\Delta}^0_1$ name for U from the $\mathbf{\Pi}^0_1$-name for B we obtain that L'_1 also has a computable right inverse.

Now let $n > 1$ and assume by induction that L_n, L'_n are computable, surjective and admit computable multi-valued right inverses, and let p be a $\mathbf{\Sigma}^0_{n+1}$-name for A, so $A = \bigcup_{\alpha\in\kappa} V_\alpha$ for $V_\alpha = \delta_{\mathbf{\Pi}^0_n(\kappa^\kappa)}(p_\alpha)$. By the induction hypothesis, we can computably find sets $A_\alpha \in \mathbf{\Delta}^0_1$ such that $L'_n(A_\alpha) = V_\alpha$. Again by Lemma 2.3.8 we can computably find a $\mathbf{\Delta}^0_1(\kappa^\kappa \times \kappa)$-name for

$$B = \bigcup_{\alpha\in\kappa} (A_\alpha \times \{\alpha\}),$$

and obtain that, if we let

$$U = \{\langle p, \alpha_1, \ldots, \alpha_{n+1} \rangle \mid (\langle p, \alpha_1, \ldots, \alpha_n \rangle, \alpha_{n+1}) \in B\},$$

then $U \in \mathbf{\Delta}^0_1(\kappa^\kappa)$ and $L_{n+1}(U) = A$. □

2.3.3 The Completeness Theorem

We finally introduce the family $(C_n)_{n\in\omega}$ and we prove the Completeness Theorem we anticipated. The proof of the Completeness Theorem essentially repeats Brattka's construction in [5, Theorem 5.5].

Definition 2.3.11 (Choice functions on κ^κ)**.** *For all $n \in \omega$, define $C_n \colon \kappa^\kappa \to \kappa^\kappa$ as*

$$C_n(p)(\alpha) = \begin{cases} 0 & \text{if } \exists \alpha_n \forall \alpha_{n-1} \ldots p(\ulcorner \alpha, \alpha_1, \ldots, \alpha_n \urcorner) \neq 0 \\ 1 & \text{otherwise} \end{cases}$$

where quantifiers are alternating and the innermost is an existential if n is odd and a universal if n is even.

Theorem 2.3.12 (Completeness Theorem)**.** *Let $n \geq 1$ and $f :\subseteq \kappa^\kappa \to \kappa^\kappa$, then*

(a) $f \leq_{\mathrm{tW}} C_n$ if and only if f is $\mathbf{\Sigma}^0_{n+1}$-measurable,

(b) assuming Hypothesis 1.2.19, $f \leq_{\mathrm{W}} C_n$ if and only if f is $\mathbf{\Sigma}^0_{n+1}$-computable.

Proof. We can show the forward direction of the theorem for both the measurable and the computable case at once by showing that C_n is $\mathbf{\Sigma}^0_{n+1}$-computable for all $n \in \omega$ and then appealing to Proposition 2.3.7.

To see this let $k \geq 1$ and consider the clopen set $A_\alpha = \{p \in \kappa^\kappa \mid p(\alpha) \neq 0\}$ and the computable function $f \colon \kappa^\kappa \to \kappa^\kappa$ defined by

$$f(p)(\alpha) = p(\ulcorner \alpha, p(0), \ldots, p(k-1) \urcorner + k).$$

The set $B_\alpha = f^{-1}[A_\alpha]$ is clopen because f is computable, and we can compute one of its clopen names from any clopen name of A_α. By Proposition 2.3.10 we can then compute a $\mathbf{\Sigma}^0_{k+1}$-name for the set

$$R_\alpha = \{p \in \kappa^\kappa \mid \exists \alpha_k, \forall \alpha_{k-1}, \ldots \langle p, \alpha_k, \ldots, \alpha_1 \rangle \in B_\alpha\},$$

and since we can compute a clopen name for A_α uniformly in α, it follows that $(R_\alpha)_{\alpha \in \kappa}$ is a computable sequence. Notice that by definition $p \in R_\alpha$ if and only if $\exists \alpha_k \forall \alpha_{k-1} \ldots (\langle p, \alpha_k, \ldots, \alpha_1 \rangle \in B_\alpha)$, if and only if

$$\exists \alpha_k \forall \alpha_{k-1} \ldots \big(f(\langle p, \alpha_k, \ldots, \alpha_1 \rangle)(\alpha) = \langle p, \alpha_k, \ldots, \alpha_1 \rangle(\ulcorner \alpha, \alpha_1, \ldots, \alpha_k \urcorner + k) = p(\ulcorner \alpha, \alpha_1, \ldots, \alpha_k \urcorner) \neq 0\big).$$

This implies that

$$R_\alpha = \{p \in \kappa^\kappa \mid \exists \alpha_k, \forall \alpha_{k-1}, \ldots p(\ulcorner \alpha, \alpha_1, \ldots, \alpha_k \urcorner) \neq 0\}.$$

We have that

$$C_k^{-1}([\alpha_0 \ldots \alpha_\beta]) = \bigcup_{\alpha_\gamma = 0} R_{\alpha_\gamma} \cap \bigcup_{\alpha_\gamma = 1} R^{\mathrm{c}}_{\alpha_\gamma}$$

hence the left hand side is computably in $\mathbf{\Sigma}^0_{k+1}$ by Lemma 2.1.12. Since sets of the form $[s]$ with $s \in \kappa^{<\kappa}$ form a basis for the topology on κ^κ, this proves that C_k is $\mathbf{\Sigma}_{k+1}$-computable and hence also $\mathbf{\Sigma}_{k+1}$-measurable.

Conversely let $F :\subseteq \kappa^\kappa \to \kappa^\kappa$ is $\mathbf{\Sigma}^0_{k+1}$-computable and assume that $\operatorname{ran}(F) \subseteq 2^\kappa$. This is possible without loss of generality as, if $\operatorname{ran}(F) \nsubseteq 2^\kappa$, we can consider the function $i \circ F$ where i is the function defined in Section 1.2.2 as

$$i(p) = [\![1100^{p(\alpha)}011]\!]_{\alpha \in \kappa}$$

since, by Corollary 1.3.11, $F \equiv_{\mathrm{sW}} i \circ F$. Now for every $\alpha \in \kappa$, consider the open set

$$A_{\alpha,0} = \{p \in \kappa^\kappa \mid p(\alpha) = 0\},$$

by $\mathbf{\Sigma}^0_{k+1}$-computability of F we can compute a $\mathbf{\Sigma}^0_{k+1}$-name for a set R_α such that

$$F^{-1}[A_{\alpha,0}] = R_\alpha \cap \operatorname{dom}(F),$$

and by Proposition 2.3.10 we can compute a $\mathbf{\Delta}^0_1$-name for a set P_α such that

$$R_\alpha = \{p \in \kappa^\kappa \mid \exists \alpha_{k+1} \forall \alpha_k \ldots \langle p, \alpha_1, \ldots, \alpha_{k+1} \rangle \in P_\alpha\}.$$

Finally, by Lemma 2.3.8, we can compute a $\mathbf{\Delta}^0_1(\kappa^\kappa \times \kappa)$-name of $P' = \bigcup_{\alpha \in \kappa}(P_\alpha \times \{\alpha\})$ and exploiting the fact that $\langle \cdot, \cdot \rangle$ is a computable homeomorphism between $\kappa^\kappa \times \kappa$ and κ^κ, we compute a $\mathbf{\Delta}^0_1(\kappa^\kappa)$-name for the set $P = \{\langle p, \alpha \rangle \mid p \in P_\alpha\}$. Similarly we can consider sets $A_{\alpha,1} = \{p \in \kappa^\kappa \mid p(\alpha) = 1\}$ and compute a $\mathbf{\Delta}^0_1$-name for a set Q defined from those in the same way as P is obtained from the $A_{\alpha,0}$s. Putting everything together we obtain:

$$F(p)(\alpha) = \begin{cases} 0 & \text{if } \exists \alpha_{k+1}, \forall \alpha_k \ldots \langle p, \alpha, \alpha_1, \ldots, \alpha_{k+1} \rangle \in P \\ 1 & \text{if } \exists \alpha_{k+1}, \forall \alpha_k \ldots \langle p, \alpha, \alpha_1, \ldots, \alpha_{k+1} \rangle \in Q \end{cases}$$

for all $p \in \mathrm{dom}(F)$, $\alpha \in \kappa$.

Now we can define the computable function $A :\subseteq \kappa^\kappa \times \kappa^\kappa$ as

$$A(p,q)(\alpha) = \begin{cases} 0 & \text{if } \exists \beta p(\ulcorner \alpha, \beta \urcorner) = 1 \\ 1 & \text{if } \exists \beta q(\ulcorner \alpha, \beta \urcorner) = 1 \end{cases}$$

with $\mathrm{dom}(A) = \{(p,q) \in \kappa^\kappa \times \kappa^\kappa \mid \forall \alpha((\exists \beta p(\ulcorner \alpha, \beta \urcorner) = 1) \oplus (\exists \beta p(\ulcorner \alpha, \beta \urcorner) = 1))\}$ and define the functions $B_P, B_Q :\subseteq \kappa^\kappa \to \kappa^\kappa$ as

$$B_P(p)(\ulcorner \ulcorner \alpha, \beta \urcorner, \alpha_1, \dots, \alpha_k \urcorner) = \begin{cases} 0 & \text{if } \langle p, \alpha, \alpha_1, \dots, \alpha_k, \beta \rangle \in P \\ 1 & \text{otherwise} \end{cases}$$

$$B_Q(p)(\ulcorner \ulcorner \alpha, \beta \urcorner, \alpha_1, \dots, \alpha_k \urcorner) = \begin{cases} 0 & \text{if } \langle p, \alpha, \alpha_1, \dots, \alpha_k, \beta \rangle \in Q \\ 1 & \text{otherwise} \end{cases}$$

and set $B = B_P \times B_Q$. Notice that B_P and B_Q are continuous as P and Q are $\mathbf{\Delta}^0_1$. Further, having access to $\mathbf{\Delta}^0_1$-names for P and Q makes these functions computable. We obtain that $(C_k \times C_k) \circ B(p) \in \mathrm{dom}(A)$ for all $p \in \mathrm{dom}(F)$ and moreover

$$A \circ (C_k \times C_k) \circ B(p)(\alpha) = 0 \iff F(p)(\alpha) = 0.$$

(for detailed computations, see the referenced proof and notice that everything can be carried out in the generalized context). Therefore, $F \leq_{\mathrm{c}} C_k \times C_k$.

Now to show that $C_k \times C_k \leq_{\mathrm{c}} C_k$ define functions $D \colon \kappa^\kappa \to \kappa^\kappa \times \kappa^\kappa$ given by $D(\langle p, q \rangle) = (p,q)$ and $E \colon \kappa^\kappa \times \kappa^\kappa \to \kappa^\kappa$ given by

$$E(p,q)(\ulcorner \beta, \alpha_1, \dots, \alpha_k \urcorner) = \begin{cases} p(\ulcorner \alpha, \alpha_1, \dots, \alpha_k \urcorner) & \text{if } \beta \text{ is the } \alpha\text{-th even ordinal} \\ q(\ulcorner \alpha, \alpha_1, \dots, \alpha_k \urcorner) & \text{if } \beta \text{ is the } \alpha\text{-th odd ordinal} \end{cases}$$

for all $p, q \in \kappa^\kappa$ and $\alpha, \beta, \alpha_1, \dots, \alpha_k \in \kappa$. The functions D and E are computable, and we obtain

$$C_k \times C_k(p,q) = D \circ C_k \circ E(p,q),$$

hence $C_k \times C_k \leq_{\mathrm{c}} C_k$ and consequently $F \leq_{\mathrm{c}} C_k$. Since both F and C_k are single valued functions on generalized Baire space, this is equivalent (by Corollary 2.3.5) to saying that $F \leq_{\mathrm{W}} C_k$, which proves the claim in the computable case. Note that in case F is just $\mathbf{\Sigma}^0_{k+1}$-measurable, then the $\mathbf{\Delta}^0_1$-sets P, Q still exist, the only difference is that we may not know how to compute their names. This means that functions A, B can be defined in the same way as above, and they will be continuous. This establishes $F \leq_{\mathrm{t}} C_k \times C_k$ and consequently $F \leq_{\mathrm{t}} C_k$, equivalently, again by Corollary 2.3.5, F is topologically Weihrauch reducible to C_k. □

This theorem implies in particular that, for all $n \in \omega$, there are $\mathbf{\Sigma}^0_{n+1}$-computable functions on κ^κ which are not $\mathbf{\Sigma}^0_n$-computable, thus providing the effective analogue of Corollary 1.1.28 for the finite levels of the $(\mathrm{Meas}_\alpha(\kappa^\kappa))_{\alpha \in \kappa^+}$ hierarchy.

Corollary 2.3.13. *For all $n \in \omega$, the function C_n is not $\mathbf{\Sigma}^0_n$-computable, so for every $n \in \omega$, there are $\mathbf{\Sigma}^0_{n+1}$-computable functions on generalized Baire space κ^κ which are not $\mathbf{\Sigma}^0_n$-computable.*

Proof. By contradiction, assume that there exists n such that C_n is $\mathbf{\Sigma}^0_n$-computable, then a fortiori C_n is $\mathbf{\Sigma}^0_n$-measurable. By Theorem 2.3.12 we know that $f \leq_{\mathrm{tW}} C_n$ for every $f \in \mathrm{Meas}_{n+1}(\kappa^\kappa)$, which by Lemma 2.3.7 implies that $f \in \mathrm{Meas}_n(\kappa^\kappa)$ for every $f \in \mathrm{Meas}_{n+1}(\kappa^\kappa)$. This is in contradiction with Corollary 1.1.28. □

Corollary 2.3.14. *For every $n \geq 1$, we have $C_n <_{\mathrm{tW}} C_{n+1}$ and consequently $C_n <_{\mathrm{W}} C_{n+1}$.*

Proof. Fix any $n \geq 1$. In the proof of Corollary 2.3.13 we have shown that C_{n+1} is not $\mathbf{\Sigma}^0_{n+1}$-measurable, so by Theorem 2.3.12 it follows that $C_{n+1} \not\leq_{\mathrm{tW}} C_n$, which implies in particular that $C_{n+1} \not\leq_{\mathrm{W}} C_n$. □

We mention here that the family $(C_n)_{n\in\omega}$ of [5] was generalized in a different direction by Galeotti in [16]. There, the author defined a family of functions $(C_{U^\alpha})_{\alpha\in\kappa^+}$ and proved ([16, Theorem 4.6.7]) that for every $\alpha \in \kappa^+$, C_{U^α} is complete with respect to $\leq_{\mathrm{tW}}$ in the set $\mathrm{Meas}_{\alpha+1}(\kappa^\kappa)$ of partial $\mathbf{\Sigma}^0_\alpha$-measurable functions from generalized Baire space to itself. In the thesis mentioned, Galeotti left the Open Question 4.6.9 asking whether it is the case that $C_{U^\alpha} <_{\mathrm{tW}} C_{U^\beta}$ for all $\alpha < \beta < \kappa^+$. We note that, exploiting Corollary 1.1.28, a proof analogous to Corollary 2.3.13 shows that this is indeed the case. Moreover, inspection of the definition of C_{U^n} for $n \in \omega$ quickly shows that $C_{U^n} \equiv_{\mathrm{W}} C_n$ for all $n \in \omega$. Our Completeness Theorem 2.3.12 then implies that each C_{U^n} is Weihrauch complete for the set of $\mathbf{\Sigma}^0_{n+1}$-computable partial function from Baire space to itself. Further, a quick computation shows that for every $\alpha < \beta < \kappa$, one can prove the reduction $C_{U^\alpha} \leq_{\mathrm{W}} C_{U^\beta}$ (note that this computation is not an effectivization of proof of $C_{U^\alpha} \leq_{\mathrm{tW}} C_{U^\beta}$ present in [16], as this proof goes through a Wadge completeness result).

Chapter 3

Choice and boundedness principles

In the [7], Brattka and Gherardi used their choice and boundedness principles on $\mathbb{R}$ and ω^ω (indicated by the letters C and B, respectively) to precisely classify the Weihrauch degrees of many of the cornerstones of real analysis. We introduce the generalized choice principles and related boundedness principles for $\mathbb{R}_\kappa$, as well as choice principles on κ, κ^κ and 2^κ and generalized omniscience principles. These include the principles introduced by Galeotti in [16], [17] and Galeotti and Nobrega in [19]. Similarly to their classical analogues, these generalized principles will capture the Weihrauch degrees of the few existing results in generalized analysis.

3.1 More represented spaces

We introduce some more represented spaces necessary to state the choice and boundedness principles that we will classify in Chapter 5.

Definition 3.1.1 (Bounded sequence spaces). *We call a sequence $(q_\alpha)_{\alpha\in\kappa}$ of κ-rationals* strictly increasing *(risp.* strictly increasing*) if for every $\alpha < \beta < \kappa$, it holds that $q_\alpha < q_\beta$ (risp. $q_\alpha > q_\beta$). We call a sequence of κ-rationals $(q_\alpha)_{\alpha\in\kappa}$* bounded *if there exists a positive κ-rational r such that for all α, it holds that $-q < q_\alpha < q$. We denote the set of* bounded strictly increasing *sequences of κ-rationals as $\mathbf{S}^{\mathrm{b}}_{\uparrow}$ and the set of* bounded strictly decreasing *sequences of κ-rationals as $\mathbf{S}^{\mathrm{b}}_{\downarrow}$. We define corresponding representations $\delta_{\mathbf{S}^{\mathrm{b}}_{\uparrow}}$ and $\delta_{\mathbf{S}^{\mathrm{b}}_{\downarrow}}$ as, for all $p \in \kappa^\kappa$*

$$\delta_{\mathbf{S}^{\mathrm{b}}_{\uparrow}}(p) = (q_\alpha)_{\alpha\in\kappa} \iff \text{for all } \alpha \in \kappa,\ \delta_{\mathbb{Q}_\kappa}(p_\alpha) = q_\alpha,$$

and

$$\delta_{\mathbf{S}^{\mathrm{b}}_{\downarrow}}(p) = (q_\alpha)_{\alpha\in\kappa} \iff \text{for all } \alpha \in \kappa,\ \delta_{\mathbb{Q}_\kappa}(p_\alpha) = q_\alpha.$$

Note that, in the classical context, the Dedekind completeness of the real line implies that every bounded strictly increasing sequence of rationals $(q_i)_{i\in\omega}$, can be associated with a unique real number, namely its least upper bound. It is then natural to consider the functions $\delta_{\mathbb{R}_<} :\subseteq \omega^\omega \to \mathbb{R}$ and $\delta_{\mathbb{R}_<} :\subseteq \omega^\omega \to \mathbb{R}$ given by, for every $p \in \omega^\omega$:

$$\delta_{\mathbb{R}_<}(p) = r \iff \nu_{\mathbb{Q}}(p(i)) = q_i\ \forall i \in \omega,\ \text{where } (q_i)_{i\in\omega} \text{ is strictly increasing and } \lim_{i\in\omega} q_i = r,$$

and

$$\delta_{\mathbb{R}_>}(p) = r \iff \nu_{\mathbb{Q}}(p(i)) = q_i\ \forall i \in \omega,\ \text{where } (q_i)_{i\in\omega} \text{ is strictly decreasing and } \lim_{i\in\omega} q_i = r,$$

where $\nu_{\mathbb{Q}}$ is any effective enumeration of the rationals. These functions are then representations of $\mathbb{R}$ itself. The classical boundedness principles are expressed in terms of the represented spaces $\mathbb{R}_< = (\mathbb{R}, \delta_{\mathbb{R}_<})$ and $\mathbb{R}_> = (\mathbb{R}, \delta_{\mathbb{R}_>})$. In the generalized context, we can no longer assume that strictly monotone, bounded sequences are convergent, therefore we need to resort to the sequence spaces $\mathbf{S}^{\mathrm{b}}_{\uparrow}$ and $\mathbf{S}^{\mathrm{b}}_{\downarrow}$ in order to state fully general boundedness principles. Nevertheless, we also introduce convergent sequence spaces in the generalized context.

Definition 3.1.2 (Convergent sequence spaces)**.** *We denote the set of* convergent strictly increasing *sequences of κ-rationals as* $\mathbf{S}^{\mathrm{c}}_{\uparrow}$ *and the set of* convergent strictly decreasing *sequences of κ-rationals as* $\mathbf{S}^{\mathrm{c}}_{\downarrow}$*.*

Notice that trivially $\mathbf{S}^{\mathrm{c}}_{\uparrow} \subseteq \mathbf{S}^{\mathrm{b}}_{\uparrow}$ and $\mathbf{S}^{\mathrm{c}}_{\downarrow} \subseteq \mathbf{S}^{\mathrm{b}}_{\downarrow}$. Convergent sequence spaces allow us to define two more representations of $\mathbb{R}_\kappa$

Definition 3.1.3 (Lower and upper κ-reals)**.** *We define $\delta_{(\mathbb{R}_\kappa)_<}$ and $\delta_{(\mathbb{R}_\kappa)_>}$ as, for all $p \in \kappa^\kappa$*

$$\delta_{(\mathbb{R}_\kappa)_<}(p) = r \iff \delta_{\mathbf{S}^{\mathrm{b}}_{\uparrow}}(p) = (q_\alpha)_{\alpha\in\kappa} \in \mathbf{S}^{\mathrm{c}}_{\uparrow} \wedge \lim_{\alpha\in\kappa} q_\alpha = r,$$

and similarly

$$\delta_{(\mathbb{R}_\kappa)_>}(p) = r \iff \delta_{\mathbf{S}^{\mathrm{b}}_{\downarrow}}(p) = (q_\alpha)_{\alpha\in\kappa} \in \mathbf{S}^{\mathrm{c}}_{\downarrow} \wedge \lim_{\alpha\in\kappa} q_\alpha = r.$$

These two representations yield represented spaces $(\mathbb{R}_\kappa, \delta_{(\mathbb{R}_\kappa)_>})$ and $(\mathbb{R}_\kappa, \delta_{(\mathbb{R}_\kappa)_<})$. In the rest of this thesis we will refer to these spaces as $(\mathbb{R}_\kappa)_>$ and $(\mathbb{R}_\kappa)_<$ respectively. As we will see, the fact that the spaces of strictly monotone bounded sequences of κ-rationals do not coincide with the spaces of strictly monotone convergent sequences of κ-rationals is the reason why the classical boundedness principles are split in two in the generalized context.

We now turn to the definitions of the representations of closed sets which we will employ for our choice principles.

Definition 3.1.4 (Representation of subsets of κ)**.** *Define a representation* $\delta'_{\mathbf{\Pi}^0_1(\kappa)} :\subseteq \kappa \to \mathbf{\Pi}^0_1(\kappa) \setminus \{\kappa\}$ *as, for $p \in \kappa^\kappa$:*

$$\delta'_{\mathbf{\Pi}^0_1(\kappa)}(p) = A \iff \mathrm{ran}(p) = \kappa \setminus A.$$

Notice that κ is discrete, hence $\mathbf{\Pi}^0_1(\kappa) = \mathcal{P}(\kappa)$. Moreover, we immediately get the following.

Lemma 3.1.5. *The representation $\delta'_{\mathbf{\Pi}^0_1(\kappa)}$ is computably equivalent to the representation $\delta_{\mathbf{\Pi}^0_1(\kappa)}$ that is obtained by viewing κ as a computable κ-metric space corestricted to $\mathbf{\Pi}^0_1(\kappa) \setminus \kappa$.*

Proof. Let $\kappa \neq A \subseteq \kappa$ and let $p \in \kappa^\kappa$ be a $\delta_{\mathbf{\Pi}^0_1(\kappa)}$-name for it, i.e.,

$$\kappa \setminus A = \bigcup_{\ulcorner \alpha,\beta \urcorner \in \mathrm{ran}(p)} B_\kappa(\alpha, \overline{\beta}).$$

Notice that for all $\ulcorner \alpha, \beta \urcorner \in \mathrm{ran}(p)$, there are three options for $B_\kappa(\alpha, \overline{\beta})$: if $\overline{\beta} > 1$, then $B_\kappa(\alpha, \overline{\beta}) = \kappa$; if $0 < \overline{\beta} < 1$, then $B_\kappa(\alpha, \overline{\beta}) = \{\alpha\}$ and lastly if $\overline{\beta} \leq 0$, $B_\kappa(\alpha, \overline{\beta}) = \emptyset$. Therefore we can compute a sequence $p' \in \kappa^\kappa$ as follows: while parsing p, we unpair elements of its range and check whether they code singletons $\{\alpha\}$, the empty set, or the entire κ. In the first case, we enumerate α, in the second case do not write anything and in the third case we stop parsing p and we start enumerating the entire κ. The sequence p' so obtained will be a $\delta'_{\mathbf{\Pi}^0_1(\kappa)}$-name for A. Conversely if q is a $\delta'_{\mathbf{\Pi}^0_1(\kappa)}$-name for A, we can compute q' as follows: whenever we read α on q, we enumerate the code for the open ball $B(\alpha, 1/2)$. The sequence q' so computed will be a $\delta'_{\mathbf{\Pi}^0_1(\kappa)}$-name for A. This shows that we can build Type 2 κ-computable functions which transform $\delta'_{\mathbf{\Pi}^0_1(\kappa)}$-names into $\delta_{\mathbf{\Pi}^0_1(\kappa)}$-names and viceversa, i.e., that the two representations given are equivalent. $\square$

We remark that the representation $\delta'_{\mathbf{\Pi}^0_1(\kappa)}$ is different from its classical analogue (called $\psi_- : \mathbb{N}^\mathbb{N} \to \mathcal{A}_-(\mathbb{N})$ in [7]), which is defined as $\psi_-(q) = A \iff \{n \mid n+1 \in \mathrm{ran}(p)\}$. This difference makes handling limit ordinals easier, but it is overall not substantial: we could still say that we represent a subset of κ by negative information, i.e., via an enumeration of its complement.

Definition 3.1.6 (Full representation of closed sets of κ-reals)**.** *Define the* full representation of closed sets *of κ-reals* $\delta^{\mathrm{full}}_{\mathbf{\Pi}^0_1(\mathbb{R}_\kappa)} :\subseteq \kappa^\kappa \to \mathbf{\Pi}^0_1(\mathbb{R}_\kappa)$ *as*

$$\delta^{\mathrm{full}}_{\mathbf{\Pi}^0_1(\mathbb{R}_\kappa)}(p) = A$$

if and only if p is an enumeration of the codes of all *the basic open balls contained in $\mathbb{R}_\kappa \setminus A$.*

By definition it is clear that if $\delta^{\text{full}}_{\mathbf{\Pi}^0_1(\mathbb{R}_\kappa)}(p) = A$, then $\delta_{\mathbf{\Pi}^0_1(\mathbb{R}_\kappa)}(p) = A$. So in particular we have

$$\delta_{\mathbf{\Pi}^0_1(\mathbb{R}_\kappa)} \leq_{\mathrm{c}} \delta^{\text{full}}_{\mathbf{\Pi}^0_1(\mathbb{R}_\kappa)}.$$

The reverse direction of this reduction does not hold, and this essentially follows from the fact that closed bounded intervals in $\mathbb{R}_\kappa$ are not κ-compact. The failure of this generalized version of the Heine-Borel Theorem was proven in [11, Proposition 5.7], we report the proof as we will need it for our construction.

Lemma 3.1.7. *Closed and bounded intervals in $\mathbb{R}_\kappa$ are not κ-compact.*

Proof. Let $I \subseteq \mathbb{R}_\kappa$ be a closed interval, $x < I$ and $y > I$. By the proof of Lemma 2.7 in [11] we can pick a strictly increasing sequence $s\colon \omega \to I$ such that the set of its upper bounds $B = \{b \in I \mid (s_i)_{i\in\omega} < b\}$ has coinitiality κ. We can then pick a sequence $t\colon \kappa \to B$ coinitial in B and define the cover $U = \{(x, s(i)) \mid i \in \omega\} \cup \{(t(\alpha), y) \mid \alpha \in \kappa\}$. It is clear that U has no subcover of size $<\kappa$, hence I is not κ-compact. □

We remark that by density of $\mathbb{Q}_\kappa$ in $\mathbb{R}_\kappa$, we can assume without loss of generality that, in the proof of Lemma 3.1.7, $x, y \in \mathbb{Q}_\kappa$ and also $(s_i)_{i\in\omega}$ and $(t_\alpha)_{\alpha\in\kappa}$ are sequences of κ-rationals.

Lemma 3.1.8. *The representation $\delta^{\text{full}}_{\mathbf{\Pi}^0_1(\mathbb{R}_\kappa)}$ is strictly more complex than $\delta_{\mathbf{\Pi}^0_1(\mathbb{R}_\kappa)}$, in other words, $\delta^{\text{full}}_{\mathbf{\Pi}^0_1(\mathbb{R}_\kappa)} \not\leq_{\mathrm{t}} \delta_{\mathbf{\Pi}^0_1(\mathbb{R}_\kappa)}$.*

Proof. Let $f :\subseteq \kappa^\kappa \to \kappa^\kappa$ be any continuous function such that for all $p \in \mathrm{dom}(\delta_{\mathbf{\Pi}^0_1(\mathbb{R}_\kappa)})$,

$$f(p) \in \mathrm{dom}(\delta^{\text{full}}_{\mathbf{\Pi}^0_1(\mathbb{R}_\kappa)}).$$

Let $C \subseteq [0, 1] \subseteq \mathbb{R}_\kappa$ be a closed interval without endpoints and let $I \subseteq \mathbb{R}_\kappa$ be another closed bounded interval contained in $\mathbb{R}_\kappa \setminus [0, +\infty)$. Consider sequences of κ-rationals $s\colon \omega \to I$ and $t\colon \kappa \to I$ as in the proof of Lemma 3.1.7. Let $x \in \mathbb{Q}_\kappa \setminus [0, 1]$ be such that $x < I$ and let $y \in \mathbb{Q}_\kappa \setminus [0, 1]$ be such that $I < y$.

Consider the cover O of $\mathbb{R}_\kappa \setminus C$ which includes every basic open ball (with rational endpoints) contained in $\mathbb{R}_\kappa \setminus (C \cup I)$ as well as the opens $\{(x, s(i)) \mid i \in \omega\}$ and $\{(t(\alpha), y) \mid \alpha \in \kappa\}$. Let p be a $\mathbf{\Sigma}^0_1(\mathbb{R}_\kappa)$-name for $\mathbb{R}_\kappa \setminus C$ corresponding to the cover O, then p is also a $\mathbf{\Pi}^0_1(\mathbb{R}_\kappa)$-name for C. By construction, no initial segment of p covers the closed interval I completely, in particular for every $\beta \in \kappa$ there exists $\gamma \in \kappa$ such that the opens $\{(t(\delta), y) \mid \delta \geq \gamma\}$ are not mentioned in $p\restriction\beta$. Now notice that $p \in \mathrm{dom}(f)$ by assumption. This implies that $f(p)$ is a well defined object.

Assume that $f(p)$ is $\delta^{\text{full}}_{\mathbf{\Pi}^0_1(\mathbb{R}_\kappa)}$-name for C, then there is some $\eta \in \kappa$ such that $f(p)(\eta)$ codes the rational open ball (x, y) and by continuity of f there is some β such that for every $p' \in \mathrm{dom}(f)$, if $p\restriction\beta \subseteq p$, then $f(p)\restriction(\eta + 1) \subseteq f(p')$. Consider any ordinal γ such that the opens $\{(t(\delta), y) \mid \delta \geq \gamma\}$ are not mentioned in $p\restriction\beta$ and let p' be a sequence extending $p\restriction\beta$ which codes the cover O' containing (in addition to the opens coded by $p\restriction\beta$) every basic open ball contained in $\mathbb{R}_\kappa \setminus [t(\delta + 2), t(\delta + 1)]$.

By construction it is clear that such a p' exists and that $\delta_{\mathbf{\Pi}^0_1(\mathbb{R}_\kappa)}(p') = [t(\delta + 2), t(\delta + 1)]$, but on the other hand we know that $f(p')$ enumerates the open (x, y), hence it cannot be the case that $\delta^{\text{full}}_{\mathbf{\Pi}^0_1(\mathbb{R}_\kappa)}(f(p')) = [t(\delta + 2), t(\delta + 1)]$. This shows that f cannot witness a reduction $\delta^{\text{full}}_{\mathbf{\Pi}^0_1(\mathbb{R}_\kappa)} \leq_{\mathrm{t}} \delta_{\mathbf{\Pi}^0_1(\mathbb{R}_\kappa)}$ and consequently that such reduction does not exist. □

For completeness we formally define the analogue of $\delta^{\text{full}}_{\mathbf{\Pi}^0_1(\mathbb{R}_\kappa)}$ for the closed interval $[0, 1]$.

Definition 3.1.9 (Full representation of closed subsets of the unit interval)**.** *Define the representation $\delta^{\text{full}}_{\mathbf{\Pi}^0_1([0,1])} :\subseteq \kappa^\kappa \to \mathbf{\Pi}^0_1([0, 1])$ as $\delta^{\text{full}}_{\mathbf{\Pi}^0_1([0,1])}(p) = A$ if and only if $\delta^{\text{full}}_{\mathbf{\Pi}^0_1(\mathbb{R}_\kappa)}(p) = A$.*

In the rest of this thesis we will always use the full representations when talking about closed sets of κ-reals.

We introduce a representation for the set of binary trees:

Definition 3.1.10 (Binary tree representation). *Recall that a binary tree is a set $T \subseteq 2^{<\kappa}$ closed under initial segments. We denote the set of binary trees as* Tr. *We introduce a representation $\delta_{\mathrm{Tr}} :\subseteq 2^\kappa \to \mathrm{Tr}$ given by $\delta_{\mathrm{Tr}}(p) = T$ if and only if for all $\alpha \in \kappa$, $p(\alpha) = 1 \iff \nu(\alpha) \in T$. We sometimes refer to a code for a tree T as its characteristic function.*

We conclude the section with an observation on the Cauchy representation for κ:

Lemma 3.1.11. *Let δ_κ be the Cauchy representation for κ, viewed as a computable κ-metric space as in Proposition 2.1.2. For every $p \in \mathrm{dom}(\delta_\kappa)$, $\delta_\kappa(p) = p(0)$.*

Proof. Let $p \in \mathrm{dom}(\delta_\kappa)$, by the fast convergence condition we have that $d_\kappa(p(0), p(1)) < 1$. Since d_κ is the discrete metric, this entails $p(1) = p(0)$. By Lemma 2.1.4 we obtain $\delta_\kappa(p) \in \mathrm{cl}(B_\kappa(p(1), 1/2))$ and again since the metric is discrete $\mathrm{cl}(B_\kappa(p(1), 1/2)) = \{p(1)\} = \{p(0)\}$, hence $\delta_\kappa(p) = p(0)$. □

3.2 Choice principles, boundedness principles, omniscience principles and results in generalized analysis

We introduce generalizations to the κ-context of all the principles in [7]. All the sets appearing in the definitions of this section are to be understood as represented spaces equipped with the representations previously introduced.

As we mentioned above, we represent closed subsets of $\mathbb{R}_\kappa$ and $[0,1] \subseteq \mathbb{R}_\kappa$ with their respective full representations $\delta^{\mathrm{full}}_{\mathbf{\Pi}^0_1(\mathbb{R}_\kappa)}$ and $\delta^{\mathrm{full}}_{\mathbf{\Pi}^0_1([0,1])}$, and we represent subsets of κ with the representation $\delta'_{\mathbf{\Pi}^0_1(\kappa)}$. For the spaces κ^κ and 2^κ we use the identity representations. Note that these are not the same as their metric representations introduced in Chapter 2, but they are equivalent to them by Corollary 2.3.3. With the identity representations on κ^κ and 2^κ, we can identify single-valued operations on these spaces with their realizers. Moreover, if f is a multi-valued operation on such spaces, then the realizers for f are precisely its choice functions.

We start with the so-called omniscience principles. The classical versions of these principles dates back to the beginning of constructive analysis. Their names come from the fact that both of them can be seen as instances of the law of the exluded middle, and both are incomputable.

Definition 3.2.1 (Omniscience principles). *We define the the κ-*limited principle of omniscience $\mathrm{LPO}_\kappa \colon \kappa^\kappa \to \kappa$ *as*

$$\mathrm{LPO}_\kappa(p) = \begin{cases} 0 & \text{if } \exists\alpha(p(\alpha) = 0) \\ 1 & \text{otherwise} \quad \text{otherwise} \end{cases}$$

*Further, we define the κ-*lesser limited principle of omniscience $\mathrm{LLPO}_\kappa \colon \kappa^\kappa \to \kappa$ *as*

$$\mathrm{LLPO}_\kappa(p) \ni \begin{cases} 0 & \text{if } \forall\alpha \text{ even } (p(\alpha) = 0) \\ 1 & \text{if } \forall\alpha \text{ odd } (p(\alpha) = 0) \end{cases}$$

where $\mathrm{dom}(\mathrm{LLPO}_\kappa) = \{p \in \kappa^\kappa \mid |\{\alpha \in \kappa \mid p(\alpha) \neq 0\}| \leq 1\}$.

We remain in the realm of recursion theory and introduce the following operations.

Definition 3.2.2 (Recursion theoretic principles). *We define the* choice operation on κ *as* $\mathrm{C}_\kappa :\subseteq \mathbf{\Pi}^0_1(\kappa) \rightrightarrows \kappa$

$$\mathrm{C}_\kappa(A) = \{\alpha \in \kappa \mid \alpha \in A\},$$

with $\mathrm{dom}(\mathrm{C}_\kappa) = \{A \subseteq \kappa \mid A \neq \emptyset \wedge A \neq \kappa\}$. *Further we define the function* $\mathrm{EC}_\kappa \colon \kappa^\kappa \to 2^\kappa$ *as*

$$\mathrm{EC}_\kappa(p) = \chi_{\mathrm{ran}(p)},$$

as well as the separation operation $\mathrm{Sep}_\kappa :\subseteq \kappa^\kappa \rightrightarrows 2^\kappa$ *given by*

$$\mathrm{Sep}_\kappa(\langle p, q\rangle) = \{\chi_A \mid \mathrm{ran}(p) \subseteq A \wedge \mathrm{ran}(q) \cap A = \emptyset\},$$

where $\mathrm{dom}(\mathrm{Sep}_\kappa) = \{\langle p,q\rangle \in \kappa^\kappa \mid \mathrm{ran}(p) \cap \mathrm{ran}(q) = \emptyset\}$*, the* extendible part operation $\mathrm{Ext} :\subseteq \mathrm{Tr} \to \mathrm{Tr}$ *given by*

$$\mathrm{Ext}(T) = \mathrm{ext}(T),$$

where $\mathrm{dom}(\mathrm{Ext}) = \{T \in \mathrm{Tr} \mid [T] \neq \emptyset\}$ *and lastly* κ-Weak Kőnig's Lemma $\mathrm{WKL}_\kappa :\subseteq \mathrm{Tr} \rightrightarrows 2^\kappa$ *given by*

$$\mathrm{WKL}_\kappa(T) = \{x \in 2^\kappa \mid x \in [T]\},$$

where $\mathrm{dom}(\mathrm{WKL}_\kappa) = \{T \in \mathrm{Tr} \mid [T] \neq \emptyset\}$.

We now turn to principles referring to $\mathbb{R}_\kappa$, starting with the various choice principles:

Definition 3.2.3 (Choice principles on $\mathbb{R}_\kappa$)**.** *We define the* closed choice operation $\mathrm{C_A} :\subseteq \mathbf{\Pi}^0_1(\mathbb{R}_\kappa) \rightrightarrows \mathbb{R}_\kappa$ *as*

$$\mathrm{C_A}(C) = \{x \in \mathbb{R}_\kappa \mid x \in C\},$$

where $\mathrm{dom}(\mathrm{C_A} = \mathbf{\Pi}^0_1(\mathbb{R}_\kappa) \setminus \emptyset$*. From it we obtain three more operations via restriction of the domain. We define the* proper closed interval choice principles $\mathrm{C_{I,c}}$ *and* $\mathrm{C^-_{I,c}}$*, and the* improper closed interval choice principle $\mathrm{C_{I,b}}$ *and* $\mathrm{C^-_{I,b}}$ *where*

$$\begin{aligned}
\mathrm{dom}(\mathrm{C_{I,c}}) &= \{[a,b] \mid 0 \leq a \leq b \leq 1\},\\
\mathrm{dom}(\mathrm{C^-_{I,c}}) &= \{[a,b] \mid 0 \leq a < b \leq 1\},\\
\mathrm{dom}(\mathrm{C_{I,b}}) &= \{I \subseteq [0,1] \mid I \neq \emptyset \wedge I \textit{ is a closed interval }\},\\
\mathrm{dom}(\mathrm{C^-_{I,b}}) &= \{I \subseteq [0,1] \mid I \textit{ is a closed interval with at least two points }\}.
\end{aligned}$$

Now we define the closely related boundedness principles:

Definition 3.2.4 (Boundedness principles on $\mathbb{R}_\kappa$)**.** *We define the function* $\mathrm{B}\colon (\mathbb{R}_\kappa)_< \to \mathbb{R}_\kappa$ *as* $\mathrm{B}(x) = x$ *and we define the function* $\mathrm{B_F}\colon (\mathbb{R}_\kappa)_< \rightrightarrows \mathbb{R}_\kappa$ *as* $\mathrm{B_F}(x) = \{y \in \mathbb{R}_\kappa \mid y \geq x\}$.

Further we define the operations $\mathrm{B_{I,c}} :\subseteq (\mathbb{R}_\kappa)_< \times (\mathbb{R}_\kappa)_> \rightrightarrows \mathbb{R}_\kappa$, $\mathrm{B^-_{I,c}} :\subseteq (\mathbb{R}_\kappa)_< \times (\mathbb{R}_\kappa)_> \rightrightarrows \mathbb{R}_\kappa$, $\mathrm{B_{I,b}} :\subseteq \mathbf{S}^{\mathrm{b}}_\uparrow \times \mathbf{S}^{\mathrm{b}}_\downarrow \rightrightarrows \mathbb{R}_\kappa$ *and* $\mathrm{B^-_{I,b}} :\subseteq \mathbf{S}^{\mathrm{b}}_\uparrow \times \mathbf{S}^{\mathrm{b}}_\downarrow \rightrightarrows \mathbb{R}_\kappa$ *as*

$$\begin{aligned}
\mathrm{B_{I,c}}(a,b) &= \{r \in \mathbb{R}_\kappa \mid a \leq r \leq b\},\\
\mathrm{B^-_{I,c}}(a,b) &= \{r \in \mathbb{R}_\kappa \mid a \leq r \leq b\},\\
\mathrm{B_{I,b}}((q_\alpha)_{\alpha\in\kappa}, (q'_\alpha)_{\alpha\in\kappa}) &= \{r \in \mathbb{R}_\kappa \mid (q_\alpha)_{\alpha\in\kappa} \leq r \leq (q'_\alpha)_{\alpha\in\kappa}\},\\
\mathrm{B^-_{I,b}}((q_\alpha)_{\alpha\in\kappa}, (q'_\alpha)_{\alpha\in\kappa}) &= \{r \in \mathbb{R}_\kappa \mid (q_\alpha)_{\alpha\in\kappa} \leq r \leq (q'_\alpha)_{\alpha\in\kappa}\},
\end{aligned}$$

where the domains are given by

$$\begin{aligned}
\mathrm{dom}(\mathrm{B_{I,c}}) &= \{(a,b) \in (\mathbb{R}_\kappa)_< \times (\mathbb{R}_\kappa)_> \mid a \leq b\},\\
\mathrm{dom}(\mathrm{B^-_{I,c}}) &= \{(a,b) \in (\mathbb{R}_\kappa)_< \times (\mathbb{R}_\kappa)_> \mid a < b\},\\
\mathrm{dom}(\mathrm{B_{I,b}}) &= \{(q_\alpha)_{\alpha\in\kappa}, (q'_\alpha)_{\alpha\in\kappa} \in \mathbf{S}^{\mathrm{b}}_\uparrow \times \mathbf{S}^{\mathrm{b}}_\downarrow \mid \exists r \in \mathbb{R}_\kappa((q_\alpha)_{\alpha\in\kappa} \leq r \leq (q'_\alpha)_{\alpha\in\kappa})\},\\
\mathrm{dom}(\mathrm{B^-_{I,b}}) &= \{(q_\alpha)_{\alpha\in\kappa}, (q'_\alpha)_{\alpha\in\kappa} \in \mathbf{S}^{\mathrm{b}}_\uparrow \times \mathbf{S}^{\mathrm{b}}_\downarrow \mid \exists r \in \mathbb{R}_\kappa((q_\alpha)_{\alpha\in\kappa} < r < (q'_\alpha)_{\alpha\in\kappa})\}.
\end{aligned}$$

Lastly, we define the operations $\mathrm{B^+_{I,b}} :\subseteq (\mathbb{Q}_\kappa)^\kappa \times ((\mathbb{Q}_\kappa) \cup \{+\infty\})^\kappa \rightrightarrows \mathbb{R}_\kappa$ *and* $\mathrm{B^+_{I,c}} :\subseteq (\mathbb{Q}_\kappa)^\kappa \times ((\mathbb{Q}_\kappa) \cup \{+\infty\})^\kappa \rightrightarrows \mathbb{R}_\kappa$ *as*

$$\mathrm{B^+_{I,b}}((q_\alpha)_{\alpha\in\kappa}, (q'_\alpha)_{\alpha\in\kappa}) = \{r \in \mathbb{R}_\kappa \mid (q_\alpha)_{\alpha\in\kappa} \leq r \leq (q'_\alpha)_{\alpha\in\kappa}\},$$

where $\mathrm{dom}(\mathrm{B^+_{I,b}})$ *is given by pairs comprised of a strictly increasing bounded sequence of* κ*-rationals* $(q_\alpha)_{\alpha\in\kappa}$ *and a strictly decreasing bounded*[1] *sequence* $(q'_\alpha)_{\alpha\in\kappa}$ *of elements in* $\mathbb{Q}_\kappa \cup \{+\infty\}$ *such that there are* κ*-reals* r *such that* $(q_\alpha)_{\alpha\in\kappa} \leq r \leq (q'_\alpha)_{\alpha\in\kappa}$ *and*

$$\mathrm{B^+_{I,c}}((q_\alpha)_{\alpha\in\kappa}, (q'_\alpha)_{\alpha\in\kappa}) = \{r \in \mathbb{R}_\kappa \mid (q_\alpha)_{\alpha\in\kappa} \leq r \leq (q'_\alpha)_{\alpha\in\kappa}\},$$

[1] In line with the use of the term in the classical context (see, e.g., page 12 in [7]), we stretch the meaning of bounded strictly decreasing sequence to accommodate for sequences ranging in $\mathbb{Q}_\kappa \cup \{+\infty\}$: in this context, we use the term to indicate the constant $+\infty$ sequence, as well as sequences $(q_\alpha)_{\alpha\in\kappa}$ such that there exists some ordinal β with $q_\gamma = +\infty$ for all $\gamma < \beta$ and $(q_\gamma)_{\beta\leq\gamma<\kappa} \in \mathbf{S}^{\downarrow}_{\mathrm{b}}$.

where $\mathrm{dom}(\mathrm{B}^+_{\mathrm{I,c}})$ *is given by pairs comprised of a strictly increasing convergent sequence of* κ*-rationals* $(q_\alpha)_{\alpha\in\kappa}$ *and a strictly decreasing convergent*[2] *sequence* $(q'_\alpha)_{\alpha\in\kappa}$ *of elements in* $\mathbb{Q}_\kappa \cup \{+\infty\}$ *such that there are* κ*-reals* r *such that* $(q_\alpha)_{\alpha\in\kappa} \leq r \leq (q'_\alpha)_{\alpha\in\kappa}$.

We remark that despite the fact that we only defined the principles B and $\mathrm{B_F}$ for lower reals, there are corresponding principles defined on upper reals. These are obviously the same as their lower real counterparts, both conceptually and in terms of Weihrauch degree. So, we concentrate on the lower real versions. We close the section with definitions of the encoding of two generalized analysis theorems, both due to Galeotti. These are the Intermediate Value Theorem for $\mathbb{R}_\kappa$ and the κ-Baire Category Theorem.

Definition 3.2.5 (Continuous functions on the unit interval)**.** *We define the set of continuous function on the unit interval* $\mathrm{C}([0,1]) = \{f\colon [0,1] \subseteq \mathbb{R}_\kappa \to \mathbb{R}_\kappa \mid f \text{ is continuous}\}$.

Note that by admissibility of the representation $\delta_{\mathbb{R}_\kappa}$ (and of its restriction to the unit interval) we know that the representation $\delta_{[[0,1]\to\mathbb{R}_\kappa]}$ (cf. Definition 1.3.2) is a total representation of the set $\mathrm{C}([0,1])$. In the rest of the thesis, we will consider $\mathrm{C}([0,1])$ as being represented by $\delta_{[[0,1]\to\mathbb{R}_\kappa]}$.

Theorem 3.2.6 (Intermediate Value Theorem for $\mathbb{R}_\kappa$)**.** *Let* $f \in \mathrm{C}([0,1])$ *be a* κ*-continuous function and let* $r \in [f(0), f(1)]$. *There is* $c \in [0,1]$ *such that* $f(c) = r$.

Proof. See [16, Theorem 3.3.1]. □

Definition 3.2.7 (IVT_κ)**.** *We define the operation* $\mathrm{IVT}_\kappa :\subseteq \mathrm{C}([0,1]) \rightrightarrows [0,1]$ *as*

$$\mathrm{IVT}_\kappa(f) = \{r \in [0,1] \mid f(r) = 0\},$$

where $\mathrm{dom}(\mathrm{IVT}_\kappa) = \{f \in \mathrm{C}([0,1]) \mid f \ \kappa\text{-continuous} \wedge f(0)\cdot f(1) < 0\}$.

Definition 3.2.8 (BCT_κ)**.** *Let* $\mathfrak{X} = (X, d, s)$ *be any* κ*-spherically complete and Cauchy complete computable* κ*-metric space, we define* $\mathrm{BCT}_\kappa(\mathfrak{X})\colon (\mathbf{\Pi}^0_1(\mathfrak{X}))^\kappa \rightrightarrows \kappa$ *as*

$$\mathrm{BCT}_\kappa(\mathfrak{X})((A_\alpha)_{\alpha\in\kappa}) = \{\beta \mid \mathrm{int}(A_\beta) \neq \emptyset\},$$

where $\mathrm{dom}(\mathrm{BCT}_\kappa(\mathfrak{X})) = \{(A_\alpha)_{\alpha\in\kappa} \mid \bigcup_{\alpha\in\kappa} A_\alpha = X\}$.

[2]In this context, consider the constant $+\infty$ sequence to be convergent to $+\infty$.

Chapter 4

Separation techniques

In [7], the so-called "Separation Techniques" are employed to prove the non-existence of reductions between the classical choice and boundedness principles, which, as we mentioned before, form the backbone of the Brattka-Gherardi diagram. In this chapter we prove the generalization to the κ-context of those techniques, in order to use them as tools for proving non-reductions between the principles introduced in Chapter 3. We will make use of these techniques in Chapter 5, where we will build our generalized diagram. In Chapter 6 we will briefly compare the role of the generalized techniques in this thesis with the role of their classical counterparts in [7].

4.1 Turing degree invariance principle

We introduce the principle of *Turing degree invariance*, which subsumes the *computable invariance principle* and the *low invariance principle* in [7] (page 19).

Definition 4.1.1. *Let $\boldsymbol{d}$ and $\boldsymbol{e}$ be two Turing degrees and let $F :\subseteq \kappa^\kappa \to \kappa^\kappa$ be any function such that for every $p \in \mathrm{dom}(F)$, if $p \leq_{\mathrm{T}} \boldsymbol{d}$, then $F(p) \leq_{\mathrm{T}} \boldsymbol{e}$. We say that F sends inputs below $\boldsymbol{d}$ into outputs below $\boldsymbol{e}$.*

Lemma 4.1.2. *Let $f :\subseteq X \rightrightarrows Y$ and $g :\subseteq U \rightrightarrows V$ be operations on represented spaces, and let $\boldsymbol{d}$, and $\boldsymbol{e}$ be two Turing degrees with $\boldsymbol{d} \leq_{\mathrm{T}} \boldsymbol{e}$. If g has a realizer G which sends inputs below $\boldsymbol{d}$ to outputs below $\boldsymbol{e}$ and $f \leq_{\mathrm{W}} g$, then f also has a realizer F which sends inputs below $\boldsymbol{d}$ to outputs below $\boldsymbol{e}$.*

Proof. Assume $f \leq_{\mathrm{W}} g$, so let $H :\subseteq \kappa^\kappa \to \kappa^\kappa$, $K :\subseteq \kappa^\kappa \to \kappa^\kappa$ be Type 2 computable functions such that, for every realizer G' of g, the function $p \mapsto H(\langle p, G' \circ K(p)\rangle)$ realizes f. In particular this means that $p \mapsto H(\langle p, G \circ K(p)\rangle)$ realizes f. Let $q \in \mathrm{dom}(f \circ \delta_X)$ be of Turing degree $\leq_{\mathrm{T}} \boldsymbol{d}$, then, by Corollary 1.2.24 we obtain that $K(q)$ has Turing degree $\leq_{\mathrm{T}} \boldsymbol{d}$ and by assumption on G, $G(K(q))$ has Turing degree $\leq_{\mathrm{T}} \boldsymbol{e}$. Therefore, $\langle q, G(K(q))\rangle$ has Turing degree $\leq_T \boldsymbol{e} \oplus \boldsymbol{d} = \boldsymbol{e}$ (recall that we use $\cup$ to denote the join operator on κ-Turing degrees) and again applying Corollary 1.2.24 we obtain that $H(\langle q, G(K(q))\rangle)$ has Turing degree $\leq_{\mathrm{T}} \boldsymbol{e}$. This shows that the realizer F of f given by $p \mapsto H(\langle p, G \circ K(p)\rangle)$ sends inputs below $\boldsymbol{d}$ into outputs below $\boldsymbol{e}$, as desired. □

This lemma is used contrapositively to prove non-reductions. We explicitly state this as a corollary.

Corollary 4.1.3 (Turing degree invariance principle)**.** *Let $f :\subseteq X \rightrightarrows Y$ and $g :\subseteq U \rightrightarrows V$ be operations on represented spaces and let $\boldsymbol{d}$, and $\boldsymbol{e}$ be two Turing degrees with $\boldsymbol{d} \leq_{\mathrm{T}} \boldsymbol{e}$. If g has a realizer G which sends inputs below $\boldsymbol{d}$ to outputs below $\boldsymbol{e}$ and f does not, then $f \not\leq_{\mathrm{W}} g$.*

4.2 Mind change invariance principle

We introduce the principle of *mind change invariance*, which is entirely analogous to the principle of the same name found in [7] (page 19).

Lemma 4.2.1. *Let $f :\subseteq X \rightrightarrows Y$ and $g :\subseteq U \rightrightarrows V$ be operations on represented spaces, $\beta \in \kappa$ an ordinal and assume that g has a realizer G which is computable with β mind changes. If $f \leq_{\mathrm{W}} g$, then f has a realizer F which is computable with β mind changes.*

Proof. Let $H :\subseteq \kappa^\kappa \to \kappa^\kappa$, $K :\subseteq \kappa^\kappa \to \kappa^\kappa$ be Type 2 computable functions such that, for every realizer G' of g, the function $p \mapsto H(\langle p, G' \circ K(p)\rangle)$ realizes f. Consider the function F given by $p \mapsto H(\langle p, G \circ K(p)\rangle)$. We have that F is a realizer of f, moreover F is obtained from G by composition with computable functions, hence we can apply Lemma 1.2.30 and obtain that F is computable with β mind changes. □

We generalize [6, Proposition 6.6] to show that the computational power of limit κ-Turing machines increases with the number of mind changes allowed.

Proposition 4.2.2. *For every $\beta \in \kappa$, define the function $\mathrm{LPO}_\kappa^\beta \colon \kappa^\kappa \to \kappa^\kappa$ as*

$$\mathrm{LPO}_\kappa^\beta(\alpha) = \begin{cases} \mathrm{LPO}_\kappa(p_\alpha) & \text{if } \alpha < \beta \\ 1 & \text{otherwise} \end{cases}$$

We have that $\mathrm{LPO}_\kappa^\beta$ is computable with β mind changes but not computable with α mind changes for any $\alpha < \beta$.

Proof. To compute $\mathrm{LPO}_\kappa^\beta$ with β mind changes, on input p we simply start printing a sequence of 1s while parsing p. For every $\alpha < \beta$, if we find a 0 on the sequence p_α (and this has not happened yet), we copy our current output, taking care to modify the α-th entry to 0, to a scratch tape T_s. Subsequently we erase the contents of the output tape, we copy the content of T_s up to β on the output tape and we continue printing 1s. The procedure described computes $\mathrm{LPO}_\kappa^\beta(p)$, and clearly needs at most β mind changes. To prove that this is optimal, given any limit machine M computing $\mathrm{LPO}_\kappa^\beta$, we recursively build a sequence $(p^\alpha)_{\alpha\in\beta} \subseteq \kappa^\kappa$ and a sequence of ordinals $(\eta_\alpha)_{\alpha\in\beta}$ such that for all $\alpha \in \beta$,

$$\mathrm{LPO}_\kappa^\beta(p^\alpha) = 0^\alpha 1^\kappa,$$

the machine M does not make any mind change in the computation of p^α after time step η_α, it prints the first β bits of output before time step η_α, and for all α and all γ with $\alpha < \gamma$, $p^\alpha{\restriction}\eta_\alpha = p^\gamma{\restriction}\eta_\alpha$. Lastly M needs at least α mind changes to compute $\mathrm{LPO}_\kappa^\beta(p^\alpha)$.

For $\alpha = 1$, consider the sequence $p^0 = 1^\kappa$ and assume that M makes γ mind changes to compute $\mathrm{LPO}_\kappa^\beta(p^0)$. Let η_1 be an ordinal such that M does not make any mind changes on input p^0 after time step η_1, and define p^1 as

$$p^1(\xi) = \begin{cases} 0 & \text{if } \xi = \ulcorner 0, \eta_1 \urcorner, \\ p^0(\xi) & \text{otherwise.} \end{cases}$$

Clearly the computation of M on p^1 coincides with the computation of M on p^0 up to step $\eta_1 + 1$, hence M makes γ mind changes before that time step, and produces a partial output $t \in \kappa^{<\kappa}$ with $t(0) = 1$. Since $\mathrm{LPO}_\kappa^\beta(p^1)(0) = 0$, M must make at least one more mind change in order to produce the correct ouput, therefore $\mathrm{mc}(M, p^1) \geq \gamma + 1 \geq 1$.

Now let $1 < \delta \leq \beta$ and inductively assume that the sequences $(p^\alpha)_{\alpha\in\delta}$, $(\eta_\alpha)_{\alpha\in\delta}$ have been defined and they enjoy the properties expressed above. By regularity of κ, let η'_δ be an ordinal in κ such that $\eta'_\delta > \sup_{\alpha\in\delta} \eta_\alpha$ and define p'^δ as

$$p'^\delta(\xi) = p^\gamma(\xi),$$

where $\gamma = \min\{\lambda \in \delta \mid (p^\sigma(\xi) = p^{\sigma'}(\xi))\, \forall \sigma, \sigma' \in [\lambda, \delta)\}$. It is clear that for all $\alpha \in \delta$ $p^\alpha{\restriction}\eta_\alpha = p'^\delta{\restriction}\eta_\alpha$. This implies at once that M makes at least $\sup_{\alpha\in\delta} \mathrm{mc}(M, p^\alpha) \geq \delta$ mind changes on p'^δ up to step η'_δ, moreover it is clear that $\mathrm{LPO}_\kappa^\beta(p'^\delta) = 0^\delta 1^\kappa$. Now let $\eta_\delta > \eta'_\delta$ be an ordinal such that the machine M has β bits written on the output tape by step η_δ and define

$$p^\delta(\xi) = \begin{cases} 0 & \text{if } \xi = \ulcorner \delta, \eta_\delta \urcorner, \\ p'^\delta(\xi) & \text{otherwise.} \end{cases}$$

Since $\mathrm{LPO}_\kappa^\beta p(\delta) = 0^{\delta+1} 1^\kappa$, we have that M needs one more mind change on p^δ than the mind changes it needed on p'^δ, hence $\mathrm{mc}(M, p^\delta) \geq \delta+1$. Moreover it is immediate to see that the sequences $(p^\alpha)_{\alpha\in\delta+1}$

and $(\eta_\alpha)_{\alpha\in\delta+1}$ are as required. This shows that the sequence $(p^\alpha)_{\alpha\in\beta}$ described above can be defined, and clearly $\sup_{\alpha\in\beta}\{\mathrm{mc}(M,p^\alpha)\} \geq \mathrm{mc}(M,p^\beta) \geq \beta$. It follows that M needs at least β mind changes to compute $\mathrm{LPO}_\kappa^\beta$, which is what we wanted to show. □

4.3 Parallelization principle

4.3.1 Introduction

We conclude the chapter with a proof of the parallelization principle [7, Lemma 4.1] in the generalized context. This is achieved under the extra assumption that κ is weakly compact, as we need the κ-compactness of generalized Cantor space 2^κ to carry out our constructions. In particular, we will prove that 2^κ is computably κ-compact relative to a so-called tree oracle. The proof of this fact draws from the papers [9] and [8], where the authors introduce the notion of computable compactness and give several sufficient conditions for it. The parallelization principle is based on the fact that $\prod \mathrm{LPO}_\kappa \not\leq_{\mathrm{tW}} \prod \mathrm{LLPO}_\kappa$. The proof we present here follows closely the proof of the same fact in the classical context, which can be found in [6, Sections 6 and 7].

First we settle a notational convention that will make our proofs easier to read.

Convention 4.3.1. *In the rest of this thesis, we will use the names* $\widehat{\mathrm{LPO}}_\kappa$ *and* $\widehat{\mathrm{LLPO}}_\kappa$ *to refer to the operations* $\prod \mathrm{LPO}_\kappa \circ \langle\cdot\rangle$ *and* $\prod \mathrm{LLPO}_\kappa \circ \langle\cdot\rangle$ *respectively, where* $\langle\cdot\rangle\colon (\kappa^\kappa)^\kappa \to \kappa^\kappa$ *is the interleaving function defined in Section 1.2.2.*

Notice that, if we represent κ^κ with the identity representation and $(\kappa^\kappa)^\kappa$ with the associated sequence space representation (cf. Definition 1.3.2), then the identity on generlized Baire space κ^κ is a realizer for $\langle\cdot\rangle$ and consequently it is also a realizer for $\langle\cdot\rangle^{-1}$. By Lemma 1.3.11 we obtain that $\prod \mathrm{LPO}_\kappa \circ \langle\cdot\rangle = \widehat{\mathrm{LPO}}_\kappa \equiv_{\mathrm{W}} \prod \mathrm{LPO}_\kappa$ and similarly $\prod \mathrm{LLPO}_\kappa \circ \langle\cdot\rangle = \widehat{\mathrm{LLPO}}_\kappa \equiv_{\mathrm{W}} \prod \mathrm{LLPO}_\kappa$, therefore we can study the degrees of the parallelized principles of omniscience via the functions $\widehat{\mathrm{LPO}}_\kappa$ and $\widehat{\mathrm{LLPO}}_\kappa$.

We now prove the obvious reduction:

Proposition 4.3.2. *We have that* $\mathrm{LLPO}_\kappa \leq_{\mathrm{sW}} \mathrm{LPO}_\kappa$

Proof. Let $G\colon \kappa^\kappa \to \kappa^\kappa$ be the Type 2 computable function given by $G(p)(\alpha) = 1 \dot{-} p(\gamma)$, where α is the γ-th even ordinal. Moreover let $H\colon \kappa \to \kappa$ be the computable function given by $H(\alpha) = 1$ if $\alpha = 0$ and $H(\alpha) = 0$ otherwise. Now let F be a realizer of LPO_κ, $p \in \mathrm{dom}(\mathrm{LLPO}_\kappa)$. We have that

$$F(G(p)) = 0 \iff \exists\alpha(G(p)(\alpha) = 0) \iff \exists\gamma(\gamma \text{ even}) \wedge (p(\gamma) \neq 0) \iff \mathrm{LLPO}_\kappa(p) \ni 1$$

and

$$F(G(p)) = 1 \iff \forall\alpha(G(p)(\alpha) \neq 0) \iff \forall\beta(\beta \text{ even}) \to (p(\beta) = 0) \iff \mathrm{LLPO}_\kappa(p) \ni 0$$

hence it follows that $H \circ F \circ G$ is a realizer of LLPO_κ. □

By monotonicity of the parallelization operator, this implies that $\widehat{\mathrm{LLPO}}_\kappa \leq_{\mathrm{W}} \widehat{\mathrm{LPO}}_\kappa$. We mention here that a straightforward generalisation of the proof of [40, Theorem 4.2] yields $\mathrm{LLPO}_\kappa <_{\mathrm{tW}} \mathrm{LPO}_\kappa$ and consequently $\mathrm{LLPO}_\kappa <_{\mathrm{W}} \mathrm{LPO}_\kappa$.

To rule out the existence of a reduction witnessing $\widehat{\mathrm{LPO}}_\kappa \leq_{\mathrm{W}} \widehat{\mathrm{LLPO}}_\kappa$, we will need several steps. The high level structure of the proof is the following: we will show that the topological Weihrauch degrees under $\widehat{\mathrm{LLPO}}_\kappa$ are closed under composition and subsequently we will show that $\widehat{\mathrm{LPO}}_\kappa \circ \widehat{\mathrm{LPO}}_\kappa \not\leq_{\mathrm{tW}} \widehat{\mathrm{LPO}}_\kappa$. This result immediately implies that $\widehat{\mathrm{LPO}}_\kappa \not\leq_{\mathrm{tW}} \widehat{\mathrm{LLPO}}_\kappa$ and consequently it excludes the possibility of a Weihrauch reduction as well. First, we show that both $\widehat{\mathrm{LLPO}}_\kappa$ and $\widehat{\mathrm{LPO}}_\kappa$ are cylinders.

Proposition 4.3.3. *We have* $\widehat{\mathrm{LPO}} \equiv_{\mathrm{sW}} \mathrm{id}_{\kappa^\kappa} \times \widehat{\mathrm{LPO}}_\kappa$ *and* $\widehat{\mathrm{LLPO}}_\kappa \equiv_{\mathrm{sW}} \mathrm{id}_{\kappa^\kappa} \times \widehat{\mathrm{LLPO}}$

Proof. By Corollary 1.3.22 it suffices to show that $\mathrm{id}_{\kappa^\kappa} \leq_{\mathrm{sW}} \widehat{\mathrm{LLPO}}_\kappa$ and $\mathrm{id}_{\kappa^\kappa} \leq_{\mathrm{sW}} \widehat{\mathrm{LPO}}_\kappa$. These reductions are proved in the same way as the corresponding classical reductions, see [6, Proposition 6.5]. □

Consequently, by Corollary 1.3.16 we have that for every operation f, $f \leq_{\mathrm{W}} \widehat{\mathrm{LLPO}}_\kappa$ if and only if $f \leq_{\mathrm{sW}} \widehat{\mathrm{LLPO}}_\kappa$ and similarly for $\widehat{\mathrm{LPO}}_\kappa$.

4.3.2 Computable compactness of generalized Cantor space and computable moduli of uniform continuity

Now we turn to the proof of $\widehat{\mathrm{LPO}}_\kappa \not\leq_{\mathrm{W}} \widehat{\mathrm{LLPO}}_\kappa$, under the extra assumption of weak compactness of κ. This first subsection is devoted to developing *computable compactness* results for generalized Cantor space 2^κ for weakly compact κ. These results yield Corollary 4.3.10, which is one of the two tools necessary for the proof of Theorem 4.3.19, which in turn yields a relatively straightforward proof $\widehat{\mathrm{LPO}}_\kappa \not\leq_{\mathrm{W}} \widehat{\mathrm{LLPO}}_\kappa$.

We introduce a computational tool which will be useful for the rest of the section.

Definition 4.3.4 (Tree oracle)**.** *We define a* tree oracle *as a function $o\colon \kappa \to \kappa$ such that for all $\gamma \in \kappa$, we have $2^{\leq\gamma} \subseteq \mathrm{ran}(\nu\restriction o(\gamma))$.*

The name tree oracle comes from the fact that, if we consider the full binary tree $2^{<\kappa}$, for any $\delta \in \kappa$ and any tree oracle o, $o(\delta)$ is an upper bound for the codes of nodes in $2^{\leq\delta}$, in the sense that (the code for) every node in $2^{\leq\delta}$ comes before $o(\delta)$. Notice that in the assumption of $\mathrm{V}_\kappa = \mathrm{L}_\kappa$, we have $\kappa = \beth_\kappa$, therefore we have $|2^{\leq\delta}| < \kappa$ for all $\delta \in \kappa$. By regularity of κ, this implies that for all $\delta \in \kappa$ there exists a bounded set $A_\delta \subseteq \kappa$ such that $\nu[A_\delta] = 2^{\leq\delta}$, hence tree oracles exist. On the other hand, it is obvious that for all $\delta \geq \omega$, it must be the case that $o(\delta) \geq |\delta|^+$, so by [12, Lemma 40] tree oracles cannot be computable functions.[1] The way we will use tree oracles is the following: let $P \subseteq 2^{<\kappa}$ be κ-decidable and consider the set $P' \subset \kappa$ given by

$$\gamma \in P' \iff \forall x \in \kappa^\gamma \, (x \in P).$$

Then it is clear that a machine with oracle access to a tree oracle o can decide P'. Obviously, the same holds for the property P'' defined by

$$\gamma \in P'' \iff \exists x \in \kappa^\gamma \, (x \in P).$$

Without the use of a tree oracle, the set P'' would only be semidecidable and the set P' would be co-semidecidable.

We introduce a representation for spaces of κ-compact subsets of κ-metric spaces.

Definition 4.3.5 (Compact sets representation)**.** *Let (X, d, s) be a computable κ-metric space and let $\mathcal{K}(X)$ be the collection of κ-compact subsets of X. Define a representation $\delta_{\mathcal{K}(X)} :\subseteq \kappa^\kappa \to \mathcal{K}(X)$ as*

$$\delta_{\mathcal{K}(X)}(p) = C$$

if and only if $\delta_{\mathbf{\Pi}^0_1(X)}(p_0) = C$ and that for every $w \in \kappa^{<\kappa}$, if w is an enumeration of codes for open balls such that $K \subseteq \bigcup_{\alpha\in\mathrm{dom}(w)} B_{w(\alpha)}$ (where $B_{w(\alpha)}$ is the basic open ball with code $w(\alpha)$), then there exists $\beta \in \kappa$ such that $\mathrm{ran}(p_\beta) = \mathrm{ran}(w)$*, and conversely for every $\alpha \in \kappa$,* $\mathrm{ran}(p_\alpha) = \mathrm{ran}(w')$ *for some w' coding a sequence of balls as above. Moreover we require that each of the sequences p_β features a marker which signals the cell δ_β such that* $\mathrm{ran}(p_\beta\restriction\delta_\beta) = \mathrm{ran}(p_\beta)$*.*

[1] In particular this implies that if o is a tree oracle the class of (Type 2) computable functions is a strict subset of the class of (Type 2) o-computable functions.

Intuitively a κ-compact name for a set $C \subseteq X$ is a closed name for C,[2] together with an enumeration of all open covers of C of size $<\kappa$. Note that, if κ is weakly compact, by Theorem 1.1.39 we have that 2^κ is κ-compact, and in particular, by Proposition 1.1.17, a set $C \subseteq 2^\kappa$ is κ-compact if and only if it is closed. Moreover, the κ-compactness of 2^κ implies that any continuous function on a closed subset of 2^κ is uniformly continuous (see Corollary 1.1.40).

We now show that, if κ is weakly compact and o is any tree oracle, then 2^κ is *computably κ-compact* relative to o, i.e., there exists a Type 2 o-computable function $H :\subseteq \kappa^\kappa \to \kappa^\kappa$ which transforms $\mathbf{\Pi}^0_1(2^\kappa)$-names for closed sets $C \subseteq 2^\kappa$ into $\mathcal{K}(X)$-names.

Lemma 4.3.6. *Assume Hypothesis 1.2.19 and let o be a tree oracle. The problem of determining whether a given sequence $(B_\beta)_{\beta<\delta}$ of basic open sets in 2^κ with $\delta < \kappa$ is a cover of 2^κ is Type 1 decidable relative to o.*

Proof. Given a (code for a) sequence of words $(w_\beta)_{\beta<\delta}$ corresponding to the given basic open sets $(B_\beta)_{\beta<\delta}$, a κ-Turing Machine can find some $\gamma \in \kappa$ such that $\gamma > \sup_{\beta\in\delta} |w_\beta|$. It can then start enumerating $2^{<\kappa}$ and check, for every $x \in 2^\gamma$, whether there exists some $\beta \in \delta$ such that $w_\beta \subseteq x$. This is a decidable property of $x \in 2^\gamma$, hence having access to o makes it possible to check whether it holds for all elements in 2^γ in $<\kappa$ steps. If this is the case, then the given sequence is a cover of 2^κ as for all $y \in 2^\kappa$ there exists some $x \in 2^\gamma$ with $x \subseteq y$ and by assumption there exists some $\beta \in \delta$ with $w_\beta \subseteq x$, hence $y \in \bigcup_{\beta\in\delta} B_\beta$. □

Proposition 4.3.7 (Computable κ-compactness of 2^κ)**.** *Assume Hypothesis 1.2.19, let o be a tree oracle and let κ be weakly compact. The function $G \colon \mathbf{\Pi}^0_1(2^\kappa) \to \mathcal{K}(2^\kappa)$ given by $C \mapsto C$ admits a realizer which is computable relative to o.*

Proof. Given a closed set $C \subseteq 2^\kappa$ and a code p for a sequence of words $(w_\alpha)_{\alpha\in\kappa}$ such that $2^\kappa \setminus C = \bigcup_{\alpha\in\kappa}[w_\alpha]$ (so essentially a $\mathbf{\Pi}^0_1$-name for C) and given a word $w \in \kappa^{<\kappa}$ coding a sequence of words $(u_\beta)_{\beta\in\delta}$ for some $\delta < \kappa$, we have that

$$C \subseteq \bigcup_{\beta\in\delta}[u_\beta] \iff \exists\gamma \left(2^\kappa \subseteq \bigcup_{\beta\in\delta}[u_\beta] \cup \bigcup_{\alpha\in\gamma}[w_\alpha] \right).$$

By Lemma 4.3.6, given γ, the test on the right is decidable relative to o, hence a machine which tries this test with increasingly large γ will be able to semidecide whether the sequence $(u_\beta)_{\beta\in\delta}$ is a cover of C. Now a machine which, in the long run, prints every possible cover of C comprised of $<\kappa$ open balls is obtained by generating all possible short sequences of words in $\kappa^{<\kappa}$ and testing them in parallel as described above. This yields the required realizer for G. □

Following [6] (see [6, Lemma 7.9]), we show that we can compute moduli of uniform continuity for computable functions on generalized Cantor space 2^κ whenever κ is weakly compact.

Lemma 4.3.8. *Assume Hypothesis 1.2.19 and let o be a tree oracle. Let κ be a weakly compact cardinal and let $F :\subseteq 2^\kappa \to 2^\kappa$ be an o-computable function, then there exists a function*

$$M :\subseteq \mathcal{K}(2^\kappa) \rightrightarrows \kappa^\kappa,$$

with $\mathrm{dom}(M) = \{C \in \mathcal{K}(2^\kappa) \mid C \subseteq \mathrm{dom}(F)\}$ *which admits an o-computable realizer and such that every $m \in M(C)$ is a modulus of uniform continuity of F on C, i.e., for all $p \in C$ and $\alpha \in \kappa$, we have $F([p{\restriction}m(\alpha)]) \subseteq [F(p){\restriction}\alpha]$.*

Proof. First, notice that F being o-computable implies that it is continuous, and hence it is uniformly continuous on all κ-compact subsets (equivalently, in this case, closed) of its domain.

Consider the following procedure: given a $\mathcal{K}(2^\kappa)$-name p of a κ-compact $C \subseteq 2^\kappa$, start running through the ordinals under κ and computing (via ν) the corresponding words $w \in 2^{<\kappa}$. Now perform

[2] By Proposition 1.1.17 it follows that in any κ-metric space, κ-compact sets are closed.

the following operations in parallel on all words $\nu(\alpha) = w \in 2^{<\kappa}$: use o to Type 1 enumerate the set $A = \{v \in 2^{<\kappa} \mid |v| = |w| \wedge w \neq v\}$, and check whether A is a cover of C by comparing it with p (recall that p contains codes for all open covers of C consisting of fewer than κ sets). If it is, print α (the ordinal code for w). In the long run, this produces a sequence $q \in \kappa^\kappa$ coding the set $\operatorname{ran}(\nu \circ q) = W$ such that $w \in W \iff [w] \subseteq 2^\kappa \setminus C$. Notice that q is (essentially) a $\mathbf{\Pi}^0_1$-name for C which specifies every basic open contained in $2^\kappa \setminus C$ (in the terminology of Definition 3.1.6, it is a full $\mathbf{\Pi}^0_1$-name for C).

Now by Proposition 1.2.21, let $f\colon 2^{<\kappa} \to 2^{<\kappa}$ be the Type 1 o-computable monotone function such that for every $x \in \operatorname{dom}(F)$,

$$F(x) = \bigcup_{s \subseteq x} f(s),$$

and, for every $\alpha \in \kappa$, consider the set

$$M_\alpha = \{\beta \in \kappa \mid \forall w \in (2^\beta \setminus W)\, |f(w)| \geq \alpha\}.$$

Note that for every α, the uniform continuity of F on C entails that M_α is nonempty and, with oracle access to o, it can be enumerated using the machine for f and q.

To see how this set is enumerated, first define the sets $(W_\gamma)_{\gamma \in \kappa}$ as

$$W_\gamma = \{w \in W \mid \text{ the code for } w \text{ appears on } q{\restriction}\gamma\}.$$

Now, given $\delta \in \kappa$, we can use the oracle o to Type 1 enumerate 2^δ and we can run through the code q to compute (increasingly accurate) approximations of the set $2^\delta \setminus W$, given by $2^\delta \setminus W_\eta$. Notice that since $|2^\delta| < \kappa$, there must be some ordinal η such that $2^\delta \setminus W = 2^\delta \setminus W_\eta$. Lastly, given a set of the form $2^\delta \setminus W_\eta$, we can run the machine for f on all words in it and check whether all computations produce an output of length $>\alpha$. We can perform the operations mentioned above in parallel for all pairs (δ, η). So, we run the machine for f in parallel on all words in $2^\delta \setminus W_\eta$ for all pairs (δ, η) until we eventually find a pair (δ', η') such that the machine for f produces an output of length $>\alpha$ on all words in $2^{\delta'} \setminus W_{\eta'}$. This is an o-computable procedure which, on every input $\alpha \in \kappa$, produces an output $\delta \in M_\alpha$. This shows that we can Type 2 compute a sequence $m \in M(C)$. By construction it is clear that m has the desired properties. Since we can do this uniformly in the code p, this proves that the function M has an o-computable realizer. $\square$

We now turn to proving results on $\widehat{\mathrm{LLPO}}_\kappa$ which are based on these computable compactness properties of 2^κ.

Proposition 4.3.9. *Assume Hypothesis 1.2.19, let o be a tree oracle and let κ be weakly compact. The function*

$$F :\subseteq \kappa^\kappa \to \mathbf{\Pi}^0_1(2^\kappa),\ p \mapsto \widehat{\mathrm{LLPO}}_\kappa(p),$$

where $\operatorname{dom}(F) = \operatorname{dom}(\widehat{\mathrm{LLPO}}_\kappa)$ *admits an o-computable realizer.*

Proof. Given any $p \in \operatorname{dom}(\widehat{\mathrm{LLPO}}_\kappa)$, we start parsing the input until we find a 1, say, in position $\ulcorner \alpha, \beta \urcorner$. If β is even, then $x(\alpha) \neq 0$ for every $x \in \widehat{\mathrm{LLPO}}_\kappa(p)$, hence $[t] \subseteq 2^\kappa \setminus \widehat{\mathrm{LLPO}}_\kappa(p)$ for all $t \in 2^{\alpha+1}$ such that $t(\alpha) = 0$. In this case we enumerate all such words t. If β is odd, the algorithm enumerates all words s in 2^α with $s(\alpha) = 1$. This can be done in $<\kappa$ steps with oracle access to o. After the enumeration phase is done, we go on parsing p. In the long run we will print a $\mathbf{\Sigma}^0_1$-name of a set $W \subseteq 2^\kappa$ such that $2^\kappa \setminus W = \widehat{\mathrm{LLPO}}_\kappa(p)$, i.e., a $\mathbf{\Pi}^0_1$-name for $\widehat{\mathrm{LLPO}}_\kappa(p)$. This shows that F has a realizer which is computable relative to o. $\square$

We immediately obtain:

Corollary 4.3.10. *Assume Hypothesis 1.2.19 and let o be a tree oracle. Let κ be a weakly compact cardinal and let $F :\subseteq 2^\kappa \to 2^\kappa$ be an o-computable function. The function $G :\subseteq 2^\kappa \rightrightarrows \kappa^\kappa$ given by*

$$G(p) = \{m \in \kappa^\kappa \mid m \text{ is a modulus of uniform continuity for } F \text{ on } \widehat{\mathrm{LLPO}}_\kappa(p)\},$$

with $\operatorname{dom}(G) = \{p \in \operatorname{dom}(\widehat{\mathrm{LLPO}}_\kappa) \mid \widehat{\mathrm{LLPO}}_\kappa(p) \subseteq \operatorname{dom}(F)\}$ *admits an o-computable realizer.*

Proof. Follows from Proposition 4.3.9, Proposition 4.3.7 and Lemma 4.3.8. $\square$

We want to use Corollary 4.3.10 to obtain the closure under composition of the topological Weihrauch degrees below $\widehat{\mathrm{LLPO}}_\kappa$. We state an intermediate result from [6].

Lemma 4.3.11. *Assume Hypothesis 1.2.19, let o be a tree oracle and let κ be a weakly compact cardinal. There exists a computable function $F\colon \kappa^\kappa \to \kappa^\kappa$ such that $\widehat{\mathrm{LLPO}}_\kappa \circ \widehat{\mathrm{LLPO}}_\kappa = \widehat{\mathrm{LLPO}}_\kappa \circ F$, hence in particular we have $\widehat{\mathrm{LLPO}}_\kappa \circ \widehat{\mathrm{LLPO}}_\kappa \leq_{\mathrm{sW}} \widehat{\mathrm{LLPO}}_\kappa$*

Proof. See [6, Lemma 7.1] for the proof of the classical counterpart. Note that the proof can be carried over to the κ-context without modifications. $\square$

4.3.3 Computable ternary extensions

We prove that, under Hypothesis 1.2.19 so-called *ternary extensions* of partial infinitary Boolean functions $f :\subseteq 2^\gamma \to 2$, with $\gamma < \kappa$, are computable relative to any tree oracle o from an ordinal which appropriately codes f. This is the second ingredient necessary for the proof of Theorem 4.3.19, which, as we mentioned earlier, is in turn necessary to obtain the Parallelization Principle for weakly compact κ (Corollary 4.3.22).

Following Brattka and Gherardi (cf. page 23 in [6]), we introduce a representation of the space $\mathbb{T} = \{0, 1/2, 1\}$ of truth values for Kleene's ternary logic which is strictly related to LLPO_κ:

Definition 4.3.12 (Representation for $\mathbb{T}$ and translation L between 2 and $\mathbb{T}$)**.** *Let $\delta_{\mathbb{T}} :\subseteq \kappa^\kappa \to \mathbb{T}$ be given by*

$$\begin{aligned}
&\delta_{\mathbb{T}}(p) = 0 \iff p \in \mathrm{dom}(\mathrm{LLPO}_\kappa) \wedge \exists\alpha \textit{ odd } (p(\alpha) = 1),\\
&\delta_{\mathbb{T}}(p) = 1 \iff p \in \mathrm{dom}(\mathrm{LLPO}_\kappa) \wedge \exists\alpha \textit{ even } (p(\alpha) = 1),\\
&\delta_{\mathbb{T}}(p) = 1/2 \iff p = 0^\kappa.
\end{aligned}$$

Further define the map $L\colon \mathbb{T} \rightrightarrows 2$ as

$$L(0) = 0, L(1) = 1, L(1/2) = \{0, 1\}.$$

The map L is strongly Weihrauch equivalent to LLPO_κ as they share the same realizations.[3] We then define the ternary extension of any infinitary Boolean function, again following Brattka and Gherardi (cf. page 24 in [6]).

Definition 4.3.13 (Ternary extension)**.** *Given any ordinal $\gamma < \kappa$ and given $f :\subseteq 2^\gamma \to 2$, define the* ternery extension of f *as the function $f' :\subseteq \mathbb{T}^\gamma \to \mathbb{T}$ given by*

$$f'((t_\alpha)_{\alpha\in\gamma}) = L'(f[\prod_{\alpha\in\gamma} L(t_\alpha)]),$$

where $\mathrm{dom}(f') = \{(t_\alpha)_{\alpha\in\gamma} \mid \prod_{\alpha\in\gamma} L(t_\alpha) \subseteq \mathrm{dom}(f)\}$ *and $L' :\subseteq \mathcal{P}(2) \to \mathbb{T}$ is defined as*

$$L'(\{0,1\}) = 1/2,\ L'(\{0\}) = 0,\ L'(\{1\}) = 1.$$

We now want to show that given any "infinitary Boolean function" which takes $<\kappa$ bits, we can o-compute a realizer of its ternary extension. To see this more easily, we borrow terminology from infinitary logic. For any ordinal $\gamma \in \kappa$, we show that any function $f :\subseteq 2^\gamma \to 2$ corresponds to an infinitary formula written in disjunctive normal form.

We work in a propositional logic which uses a γ-sequence $(v_\alpha)_{\alpha\in\gamma}$ of distinct propositional letters. We define propositional formulas by the following recursion

(a) for any $\alpha \in \gamma$, v_α is a formula,

[3]We consider 2 as being represented by the notation $\hat{\nu} :\subseteq \kappa \to 2$ given by $\hat{\nu}(0) = 0$, $\hat{\nu}(1) = 1$ and undefined for every other ordinal.

(b) if φ is a formula, then $\neg\varphi$ is a formula,

(c) if $(\varphi_\alpha)_{\alpha<\beta}$ is a sequence of formulas with $\beta < 2^{|\gamma|}$, then $\bigvee_{\alpha<\beta} \varphi_\alpha$ and $\bigwedge_{\alpha<\beta} \varphi_\alpha$ are formulas.

We can then identify an element $x \in 2^\gamma$ with an assignment, namely stipulating that the propositional letter v_α is assigned truth value i by x if and only if $x(\alpha) = i$.

Now, given $x \in 2^\gamma$, define φ_x as

$$\varphi_x = \bigwedge_{\alpha\in\gamma} p(x(\alpha)).$$

where $p(x(\alpha)) = v_\alpha$ if $x(\alpha) = 1$ and $p(x(\alpha)) = \neg v_\alpha$ if $x_\alpha = 0$. Lastly we define Φ_f as

$$\Phi_f = \bigvee_{x\in f^{-1}[1]} \varphi_x.$$

Note that for every $f :\subseteq 2^\gamma \to 2$, the formula Φ_f is in disjunctive normal form, and in particular it is a disjunction of fewer than $2^{|\gamma|}$ formulas of the form φ_x. Now it is clear that, if we interpret a sequence $x \in \mathrm{dom}(f)$ as an assignment, we have that $f(x)$ is the truth value of Φ_f under the assignment x. We prove two lemmas necessary to obtain the computability of ternary extensions.

Lemma 4.3.14. *Let $\delta < \kappa$ and let $g_\delta\colon 2^\delta \to 2$ correspond to disjunction of size δ, i.e.,*

$$g_\delta(x) = 1 \iff \exists\alpha\, x(\alpha) = 1.$$

The ternary extension of g_δ is computable.

Proof. We have to show that the function $g'_\delta\colon \mathbb{T}^\delta \to \mathbb{T}$ given by $g'_\delta((t_\alpha)_{\alpha\in\delta}) = L'(g_\delta[\prod_{\alpha\in\delta} L(t_\alpha)])$ has a computable realizer. Consider the κ-Turing machine M_δ which behaves as follows: on input a code p for some sequence in $\mathbb{T}^\delta$, it parses the sequence p, keeping track of all the ordinals $\alpha < \delta$ such that a 1 has been found in sequence p_α, and keeps on outputting 0 until one of the following conditions is met:

(a) M has found a sequence p_α such that, for some even ordinal β, $p_\alpha(\beta) = 1$ (so p_α codes $1 \in \mathbb{T}$). In this case, M outputs 1 in the next free even position, then starts outputting 0 indefinitely (so the output of M codes $1 \in \mathbb{T}$),

(b) the previous condition has not yet been met, but M has found the symbol 1 on sequence p_α for every ordinal $\alpha < \delta$. In this case, note that it must be that every p_α has its only nonzero ordinal in an odd position, hence all these p_α's code $0 \in \mathbb{T}$. In this case, M outputs 1 in the next free odd position, then starts outputting 0 indefinitely (so the output of M codes $0 \in \mathbb{T}$),

We claim that M computes g'_δ: if we let $(t_\alpha)_{\alpha\in\delta}$ be the sequence coded by p, there are three possible options: if there exists some $\beta \in \delta$ such that $t_\beta = 1$, then $L(t_\beta) = 1$ so $y(\beta) = 1$ for every $y \in \prod_{\alpha<\delta} L(t_\alpha)$. In this case $g_\delta(y) = 1$ for all $y \in \prod_{\alpha<\delta} L(t_\alpha)$, so $L'(g_\delta(y)) = 1$ and consequently $g'_\delta((t_\alpha)_{\alpha\in\delta}) = 1$. This case corresponds to case (a) in the definition of M, so running M on p outputs a code for $1 \in \mathbb{T}$. This shows that M is correct on such codes p.

If no such β exists, then we can make a second case distinction: if, for all $\alpha \in \delta$, $t_\alpha = 0$, then obviously $g'_\delta((t_\alpha)_{\alpha\in\delta}) = 0$. Note that in this case each of the sequences p_α coding the element t_α has a 1, so by regularity of κ, there must be some index $\eta < \kappa$ such that all these 1s appear before η. Again by definition of the machine, once M has gotten to read cells of its input up to index η, it will detect the fact that 1s have been found on all sequences $(p_\alpha)_{\alpha\in\delta}$ and output a code for $0 \in \mathbb{T}$. Hence the machine behaves correctly on these inputs as well.

Lastly, if the input does not fall in either of the above cases, it must be the case that $t_\alpha \neq 1$ for every $\alpha \in \delta$, and $t_\alpha = 1/2$ for at least one index α. In this case we have that the constant 0 sequence is in $\prod_{\alpha\in\delta} L(t_\alpha)$, as well as at least one sequence containing at least one 1. This clearly implies that $g'_\delta((t_\alpha)_{\alpha\in\delta}) = 1/2$. Also in this last case, the machine behaves correctly, as by its specification, it will just continue to output 0s indefinitely, so in the long run it will output 0^κ, which is the code for the correct ouput $1/2 \in \mathbb{T}$. This shows that M_δ computes a realizer for g'_δ.

Moreover, the construction is uniform in the parameter δ, as this parameter is only used in condition (b) to determine a halting condition for M_δ. This implies that the functions (g'_δ) are actually uniformly computably realized, i.e., there exists single κ-Turing machine M which, given any parameter $\delta \in \kappa$, computes a realizer for g'_δ. $\square$

Lemma 4.3.15. *Assume Hypothesis 1.2.19. Let $\gamma < \kappa$, $x \in 2^\gamma$ and let $f_x \colon 2^\gamma \to 2$ be the function corresponding to the truth function of φ_x, i.e., $f_x(y) = 1 \iff x = y$. The ternary extension f'_x of f_x is computable using the ordinal code of x as a parameter.*

Proof. Let β be an ordinal code for x. i.e., $\nu(\beta) = x$. Consider the κ-Turing Machine N which uses β as parameter and works as follows: on input a code p for some sequence $(y_\alpha)_{\alpha\in\delta} \in \mathbb{T}^\gamma$, N parses the input while keeping track of all the ordinals $\alpha \in \gamma$ such that a symbol 1 has appeared in the sequence p_α. While doing this, it keeps outputting 0s until one of the following conditions is met:

(a) for some ordinal α, the machine finds a 1 on the sequence p_α and learns that $y_\alpha = 1 - x(\alpha)$.[4] In this case N outputs a 1 on the next even position, then proceeds to print 0s indefinitely (so the output is a code for $0 \in \mathbb{T}$),

(b) if no ordinal satisfying the above condition has been found, and N learns that, for every $\alpha \in \gamma$, the symbol 1 has been found on the sequence p_α, then N outputs a 1 on the next odd position and then proceeds to output 0s indefinitely (so the output is a code for $1 \in \mathbb{T}$).

To see that N computes f'_x we reason similarly to the proof of Lemma 4.3.14: if any y_β has value $1 - x(\beta)$, then $f'_x((y_\alpha)_{\alpha\in\gamma}) = 0$, and this corresponds to the behaviour of N thanks condition (a). If this condition is never satisfied, then either $y_\beta = x(\beta)$ for every β, in which case $f'_x(y_\alpha)_{\alpha\in\gamma}) = 1$ and N acts accordingly thanks to the second condition, or we have that $y_\beta = 1/2$ or $y_\beta = x_\beta$ for every β (with at least one β with $y_\beta = 1/2$). In this case, we have that $f'_x(y_\alpha)_{\alpha\in\gamma}) = 1/2$ and, since neither of the two conditions above is ever met, N outputs 0^κ, which is the correct code. $\square$

This implies in particular that, if $\delta \in \kappa$ and $(x_\alpha)_{\alpha\in\delta}$ is a sequence of elements of 2^γ, then the function $F' \colon \mathbb{T}^\gamma \to \mathbb{T}^\delta$ defined by $s \mapsto (f'_{x_0}(s), \dots, f'_{x_\alpha}(s), \dots)$ is computable (cf. Proposition 1.2.28), from an ordinal parameter which appropriately codes the sequence $(x_\alpha)_{\alpha\in\delta}$.

We can now prove that ternary extensions are o-computable for any tree oracle o. To do so, we will need to define an encoding of partial infinitary Boolean functions $f :\subseteq 2^\gamma \to 2$ suitable for Type 1 computations. The use of the oracle o is and sufficient to be able to code functions $f :\subseteq 2^\gamma \to 2$ for any $\gamma \in \kappa$. We spell out the coding details to show why this is the case: given an ordinal γ and a function $f :\subseteq 2^\gamma \to 2$, we code f as an ordinal η defined as

$$\eta = \nu^{-1}(s),$$

where ν^{-1} stands for a computable right inverse of ν and $s \in 2^{<\kappa}$ is a word defined as

$$\begin{aligned}
s(\ulcorner \alpha, 0, 0 \urcorner) = 1 &\iff \nu(\alpha) \in \mathrm{dom}(f) \wedge \alpha < o(\gamma), \\
s(\ulcorner \alpha, 1, 0 \urcorner) = 1 &\iff s(\ulcorner \alpha, 0, 0 \urcorner) = 1 \wedge f(\nu(\alpha)) = 0, \\
s(\ulcorner \alpha, 0, 1 \urcorner) = 0 &\iff s(\ulcorner \alpha, 0, 0 \urcorner) = 1 \wedge f(\nu(\alpha)) = 1, \\
s(\beta) = 0 &\text{ for all other } \beta \in \mathrm{dom}(s),
\end{aligned}$$

with $\mathrm{dom}(s) = \min\{\delta \in \kappa \mid \delta > \sup_{\alpha\in o(\gamma)} \ulcorner \alpha, 1, 1 \urcorner\}$. It is straightforward to see that if δ is an ordinal, deciding whether δ codes a partial infinitary Boolean function $f :\subseteq 2^\gamma \to 2$ as described above is an o-computable procedure. Moreover, whenever this is the case, we can o-computably obtain $f(x)$ for any $x \in \mathrm{dom}(f)$ from the ordinal δ. Conversely, if M is a machine computing f and which has access to a parameter δ coding $\mathrm{dom}(f)$ (note that $\chi_{\mathrm{dom}(f)}$ is itself a word $y \in 2^\gamma$, so it can be coded as $\nu^{-1}(y)$), M can be used to Type 1 compute an ordinal coding f.

[4] The machine can check the value of $x(\alpha)$ because it has access to the parameter β.

Proposition 4.3.16. *Assume Hypothesis 1.2.19 and let o be a tree oracle. Let γ be an ordinal below κ and let $f :\subseteq 2^\gamma \to 2$ be a function coded by the ordinal α. There exists a T2κTM M which uses the parameter α and computes a realization of f' the ternary extension of f.*

Proof. Given a parameter α coding $f :\subseteq 2^\gamma \to 2$, we can computably obtain a (code for) a sequence $(x_\alpha)_{\alpha\in\delta}$ such that $f^{-1}[1] = \{x_\alpha \in \mathrm{dom}(f) \mid \alpha \in \delta\}$ where $\delta < \kappa$. By Lemma 4.3.15, we can compute, for every x_α, a realizer for the function f'_{x_α} (which is actually the truth function for φ_{x_α}) and hence we can compute a realizer for the function $F' :\subseteq \mathbb{T}^\gamma \to \mathbb{T}^\delta$ given by $F'(s) = (f'_{x_\alpha}(s))_{\alpha\in\delta}$ with $\mathrm{dom}(F') = \mathrm{dom}(f')$. Now $f' = g'_\delta \circ F'$, and since g'_δ is computable by Lemma 4.3.14, f' is computable. □

Note that the constructions in Lemma 4.3.14 and 4.3.15 are uniform in the parameters δ, γ, hence Proposition 4.3.16 can be strengthened to the following (cf. [6, Corollary 7.7]):

Corollary 4.3.17. *Assume Hypothesis 1.2.19 and let o be a tree oracle. There exists a κ-Turing Machine M which uses the oracle o which, on input γ and a code for $f :\subseteq 2^\gamma \to 2$, computes a code for a T2κTM N which realizes the ternary extension f' of f.*

4.3.4 Proof of the Parallelization principle

We keep on following [6] to prove Theorem 4.3.19, which is a closure property of the Weihrauch degrees, relativized to a tree oracle o, below the degree of $\widehat{\mathrm{LLPO}}_\kappa$ (cf. [6, Theorem 7.11]). We use this result to obtain the Parallelization principle for weakly compact κ.

Proposition 4.3.18. *Assume Hypothesis 1.2.19, let o be a tree oracle, let κ be a weakly compact cardinal and let $F :\subseteq 2^\kappa \to 2^\kappa$ be computable relative to o, then there exists some o-computable $G :\subseteq 2^\kappa \to 2^\kappa$ such that*

$$F \circ \widehat{\mathrm{LLPO}}_\kappa = \widehat{\mathrm{LLPO}}_\kappa \circ G.$$

Proof. Let $p \in \mathrm{dom}(F \circ \widehat{\mathrm{LLPO}}_\kappa)$, then by definition of composition we have $\widehat{\mathrm{LLPO}}_\kappa(p) \subseteq \mathrm{dom}(F)$, hence F is uniformly continuous on $\widehat{\mathrm{LLPO}}_\kappa(p)$. By Propositions 4.3.9 and 4.3.7, we can o-compute a $\mathcal{K}(2^\kappa)$ name of $\widehat{\mathrm{LLPO}}_\kappa(p)$ and then by Lemma 4.3.8 we can o-compute a function $m_p\colon \kappa \to \kappa$ which is a modulus of uniform continuity of F on $\widehat{\mathrm{LLPO}}_\kappa(p)$. Notice that we can assume without loss of generality that $m_p(\alpha) \geq 1$ for all $\alpha \in \kappa$ (this is because we can just impose this extra condition on the computable realizer which we build in the proof of Lemma 4.3.8).

Using a machine for F, given any $\gamma \in \kappa$ we can (uniformly in γ and p) o-compute (a code for) the infinitary Boolean function $f_{p,\gamma} :\subseteq 2^{m_p(\gamma+1)+1} \to 2$ such that

$$F(q)(\gamma) = f_{p,\gamma}((q(\alpha)_{\alpha\in m_p(\gamma+1)+1}))$$

for any $q \in \widehat{\mathrm{LLPO}}_\kappa(p)$. To simplify the notation, we will denote $m_p(\gamma+1)+1$ as $\hat{\gamma}$ from this point of the proof. Note that the functions $f_{p,\gamma}$ have domains given by

$$\mathrm{dom}(f_{p,\gamma}) = \{x \in 2^{\hat{\gamma}} \mid \exists q \in \widehat{\mathrm{LLPO}}_\kappa(p)\,(x \subseteq q)\}.$$

By Corollary 4.3.17, we obtain (codes for) realizers $G_{p,\gamma} :\subseteq 2^\kappa \to 2^\kappa$ of the ternary extensions $f'_{p,\hat{\gamma}}$.

We spell out what this means for clarity. First, recall that a code p for $t \in \mathbb{T}^{\hat{\gamma}}$ is a sequence such that $\delta_{\mathbb{T}}(p_\alpha) = t_\alpha$ for all $\alpha \in \hat{\gamma}$. Now the functions $f'_{p,\gamma}$ have domains given by

$$\mathrm{dom}(f'_{p,\gamma}) = \{(t_\alpha)_{\alpha\in\hat{\gamma}} \mid \prod_{\alpha\in\hat{\gamma}} L(t_\alpha) \subseteq \mathrm{dom}(f_{p,\gamma})\}.$$

Hence the set $\mathrm{dom}(f'_{p,\gamma} \circ \delta_{\mathbb{T}^{\hat{\gamma}}})$ is a superset of $A = \{q \in \mathrm{dom}(\widehat{\mathrm{LLPO}}_\kappa) \mid \widehat{\mathrm{LLPO}}_\kappa(q) \subseteq \mathrm{dom}(f_{p,\gamma})\}$ which in turn is a superset of $\mathrm{dom}(F) \cap \widehat{\mathrm{LLPO}}_\kappa(p)$. Second, a realizer $G_{p,\gamma}$ for $f'_{p,\gamma}$ is a function such that for all sequences $q \in \mathrm{dom}(G_{p,\gamma})$ coding sequences $(t_\alpha)_{\alpha\in\hat{\gamma}} \in \mathrm{dom}(f'_{p,\gamma})$, $G_{p,\gamma}(q)$ is a code for $f'_{p,\gamma}((t_\alpha)_{\alpha\in\hat{\gamma}})$. Lastly, by definition of $\delta_{\mathbb{T}}$ we obtain that for all codes $q \in \mathrm{dom}(\delta_{\mathbb{T}})$, $\mathrm{LLPO}_\kappa(q) = L(\delta_{\mathbb{T}}(q))$.

Now we can use the codes for the realizers $G_{p,\gamma}$ to o-compute the function $G :\subseteq 2^\kappa \to 2^\kappa$ defined as

$$G(p) = \langle G_{p,0}(p), \ldots, G_{p,\alpha}(p), \ldots, \rangle.$$

By the argument above, we obtain that for all $p \in \operatorname{dom}(F \circ \widehat{\mathrm{LLPO}})$, $p \in \operatorname{dom}(\widehat{\mathrm{LLPO}}_\kappa \circ G)$ and

$$\widehat{\mathrm{LLPO}}_\kappa(G(p)) = \prod_{\alpha\in\kappa} \mathrm{LLPO}_\kappa(G_{p,\alpha}(p)) = \prod_{\alpha\in\kappa} L(\delta_{\mathbb{T}}(G_{p,\alpha}(p))).$$

Again by the argument above we have that for every $\alpha \in \kappa$, $G_{p,\alpha}(p)$ codes $f'_{p,\gamma}((t_\eta)_{\eta\in\alpha})$ where $(t_\eta)_{\eta\in\alpha} = \delta_{\mathbb{T}^\alpha}(p)$, i.e., $\mathrm{LLPO}_\kappa(p_\eta) = t_\eta$ for all $\eta \in \alpha$. Hence

$$\prod_{\alpha\in\kappa} L(\delta_{\mathbb{T}}(G_{p,\alpha}(p))) = \prod_{\alpha\in\kappa} L((f'_{p,\alpha}(\delta_{\mathbb{T}^\alpha}(p)))).$$

Notice that for any ordinal $\alpha \in \kappa$ and sequence $p \in \operatorname{dom}(\widehat{\mathrm{LLPO}}_\kappa \circ G)$, by definition we have that $L(f'_{p,\alpha}(\delta_{\mathbb{T}^\alpha}(p))) = f_{p,\alpha}[\{x \in 2^\alpha \mid \exists q \in \widehat{\mathrm{LLPO}}_\kappa(p)\,(x \subseteq q)\}]$, hence

$$\prod_{\alpha\in\kappa} L((f'_{p,\alpha}(\delta_{\mathbb{T}^\alpha}(p)))) = \prod_{\alpha\in\kappa} \{f_{p,\alpha}(x) \mid x \in 2^\alpha \wedge [x] \cap \mathrm{LLPO}_\kappa(p) \neq \emptyset\}.$$

By definition the functions $f_{p,\gamma}$, for all $x \in \operatorname{dom}(f_{p,\gamma})$ and all $p \in \operatorname{dom}(F \circ \widehat{\mathrm{LLPO}}_\kappa)$ with $x \subseteq q$ for some $q \in \widehat{\mathrm{LLPO}}_\kappa(p)$ we have $F(q)(\gamma) = f_{p,\gamma}(x)$, hence we finally obtain

$$\prod_{\alpha\in\kappa} \{f_{p,\alpha}(x) \mid x \in 2^\alpha \wedge [x] \cap \mathrm{LLPO}_\kappa(p) \neq \emptyset\} = F[\widehat{\mathrm{LLPO}}_\kappa(p)].$$

Putting everything together, this shows that $\widehat{\mathrm{LLPO}}_\kappa \circ G = F \circ \widehat{\mathrm{LLPO}}_\kappa$ thus proving the claim. □

We remark that this result is the generalized analogue of [6, Theorem 7.10].

Finally we can prove that operations which Weihrauch reduce to $\widehat{\mathrm{LLPO}}_\kappa$ relative the oracle o are closed under composition:

Theorem 4.3.19. *Assume Hypothesis 1.2.19, let o be a tree oracle, let κ be a weakly compact cardinal, and let f and g be two composable operations on represented spaces such that $f \leq^o_{\mathrm{W}} \widehat{\mathrm{LLPO}}_\kappa$ and $g \leq^o_{\mathrm{W}} \widehat{\mathrm{LLPO}}_\kappa$, then $g \circ f \leq^o_{\mathrm{W}} \widehat{\mathrm{LLPO}}_\kappa$.*

Proof. Since $\widehat{\mathrm{LLPO}}_\kappa$ is a cylinder, we can assume both reductions are actually strong reductions (see Corollary 1.3.16), hence there are o-computable functions H, K, H', K' such that for every L realizing $\widehat{\mathrm{LLPO}}_\kappa$, $H' \circ L \circ K'$ is a realizer of f and $H \circ L \circ K$ is a realizer of g. Therefore, for any pair L and L' of realizers of $\widehat{\mathrm{LLPO}}_\kappa$, we have that $H \circ L \circ K \circ H' \circ L' \circ K'$ is a realizer of $g \circ f$. Since we can assume that $\operatorname{dom}(K \circ H') \cup \operatorname{ran}(K \circ H') \subseteq 2^\kappa$, it follows by Proposition 4.3.18 that there exists some o-computable function G such that $K \circ H' \circ \widehat{\mathrm{LLPO}}_\kappa = \widehat{\mathrm{LLPO}}_\kappa \circ G$. Let F be the computable function of Lemma 4.3.11 such that $\widehat{\mathrm{LLPO}}_\kappa \circ \widehat{\mathrm{LLPO}}_\kappa = \widehat{\mathrm{LLPO}}_\kappa \circ F$ and define $K'' = F \circ G \circ K'$. We obtain

$$H \circ \widehat{\mathrm{LLPO}}_\kappa \circ K \circ H' \circ \widehat{\mathrm{LLPO}}_\kappa \circ K' = H \circ \widehat{\mathrm{LLPO}}_\kappa \circ K''$$

which in particular implies that for any realizer L of $\widehat{\mathrm{LLPO}}_\kappa$, $H \circ L \circ K''$ realizes $g \circ f$, i.e.. $g \circ f \leq^o_{\mathrm{W}} \widehat{\mathrm{LLPO}}_\kappa$. □

Similarly to the classical analogue, this result relativizes to any oracle which can compute o. Now, by Proposition 1.2.22, for any operation f, if $f \leq_{\mathrm{tW}} \widehat{\mathrm{LLPO}}_\kappa$, then there exists an oracle q such that $f \leq^q_{\mathrm{W}} \widehat{\mathrm{LLPO}}_\kappa$. Without loss of generality we can assume that $o \leq_{\mathrm{T}} q$, as if this is not the case we can consider $q' = \langle o, q\rangle$. Therefore if $f \leq_{\mathrm{tW}} \widehat{\mathrm{LLPO}}_\kappa$ and $g \leq_{\mathrm{tW}} \widehat{\mathrm{LLPO}}_\kappa$, there exists some oracle p with $o \leq_{\mathrm{T}} p$ such that $f \leq^p_{\mathrm{W}} \widehat{\mathrm{LLPO}}_\kappa$ and $g \leq^p_{\mathrm{W}} \widehat{\mathrm{LLPO}}_\kappa$. This implies that we can appeal to Theorem 4.3.19 and obtain that $g \circ f \leq^p_{\mathrm{W}} \widehat{\mathrm{LLPO}}_\kappa$ and in particular $g \circ f \leq_{\mathrm{tW}} \widehat{\mathrm{LLPO}}_\kappa$. This shows that the topological Weihrauch degrees under $\widehat{\mathrm{LLPO}}_\kappa$ are also closed under composition. The situation is different for $\widehat{\mathrm{LPO}}_\kappa$:

Lemma 4.3.20. *We have* $\widehat{\mathrm{LPO}}_\kappa \circ \widehat{\mathrm{LPO}}_\kappa \not\leq_{\mathrm{tW}} \widehat{\mathrm{LPO}}_\kappa$.

Proof. As in the classical case (see [6, Lemma 7.2]), a simple calculation shows that $\widehat{\mathrm{LPO}}_\kappa = C_1$ and $\widehat{\mathrm{LPO}}_\kappa \circ \widehat{\mathrm{LPO}}_\kappa = C_2$. By Corollary 2.3.14 we obtain $\widehat{\mathrm{LPO}}_\kappa \circ \widehat{\mathrm{LPO}}_\kappa \not\leq_{\mathrm{tW}} \widehat{\mathrm{LPO}}_\kappa$. □

This immediately implies:

Corollary 4.3.21. *Assuming Hypothesis 1.2.19, if* κ *is weakly compact, then* $\widehat{\mathrm{LPO}}_\kappa \not\leq_{\mathrm{tW}} \widehat{\mathrm{LLPO}}_\kappa$.

Similarly to the other separation techniques, this result is used contrapositively to obtain the so-called parallelization principle. We state it here, with explicit mention of the assumption of weak compactness of κ.

Corollary 4.3.22 (Parallelization Principle)**.** *Assume Hypothesis 1.2.19, let* κ *be a weakly compact cardinal and let* f *and* g *be operations on representd spaces such that* $\mathrm{LPO}_\kappa \leq_{\mathrm{W}} f$ *and* $g \leq_{\mathrm{W}} \widehat{\mathrm{LLPO}}_\kappa$. *We have* $f \not\leq_{\mathrm{W}} g$.

Proof. We prove the claim by contradiction: assume $f \leq_{\mathrm{W}} g$, then by transitivity we obtain $\mathrm{LPO}_\kappa \leq_{\mathrm{W}} \widehat{\mathrm{LLPO}}_\kappa$ and by the monotonicity of parallelization (cf. Proposition 1.3.18) we obtain $\widehat{\mathrm{LPO}}_\kappa \leq_{\mathrm{W}} \widehat{(\widehat{\mathrm{LLPO}}_\kappa)}$, By idempotency of parallelization we have $\widehat{(\widehat{\mathrm{LLPO}}_\kappa)} \equiv_{\mathrm{W}} \widehat{\mathrm{LLPO}}_\kappa$, so in the end we obtain $\widehat{\mathrm{LPO}}_\kappa \leq_{\mathrm{W}} \widehat{\mathrm{LLPO}}_\kappa$. This is in contradiction with Corollary 4.3.21. □

Chapter 5

Classification Results

The following chapter contains results concerning the Weihrauch degrees of the principles introduced in Chapter 3. As we said before, the original aim of this thesis was to obtain a classification analogous to the one in [7] for our generalized principles. A study of these generalized principles was started by Galeotti in [16], where the author focused on topological Weihrauch degrees, and was expanded by Galeotti in [17] and Galeotti and Nobrega in [19]. We interleave the exposition of our results with discussion of how the latter compare with their classical analogues. Whenever unspecified, assume that the classical spaces are equipped with representations analogous to the representations introduced in Chapters 2 and 3 for their generalized counterparts.

5.1 What we have shown so far

We start the chapter with a proposition summarising the classification results already proven earlier in this thesis.

Proposition 5.1.1. *The following relations hold:*

(a) assuming Hypothesis 1.2.19, for every $n \geq 1$, $C_n <_{\mathrm{W}} C_{n+1}$ and C_n is complete with respect to $\leq_{\mathrm{W}}$ for the set of single-valued, partial, $\mathbf{\Sigma}^0_{n+1}$-effectively measurable functions on κ^κ,

(b) $\widehat{\mathrm{LPO}}_\kappa \equiv_{\mathrm{W}} C_1$,

(c) $\mathrm{LLPO}_\kappa <_{\mathrm{W}} \mathrm{LPO}_\kappa$, hence also $\widehat{\mathrm{LLPO}}_\kappa \leq_{\mathrm{W}} \widehat{\mathrm{LPO}}_\kappa$,

(d) assuming Hypothesis 1.2.19 and weak compactness of κ, $\widehat{\mathrm{LLPO}}_\kappa <_{\mathrm{W}} \widehat{\mathrm{LPO}}_\kappa$,

(e) if f is a computable operation between represented spaces, $\mathrm{LLPO}_\kappa \not\leq_{\mathrm{W}} f$.

Proof. Item (a) is Theorem 2.3.12 and Corollary 2.3.14. Item (b) is trivial as $\widehat{\mathrm{LPO}}_\kappa$ and C_1 are actually the same function. The positive part of item (c) is in Proposition 4.3.2, the negative part is, as we mentioned in Section 4.3, a consequence of the straightforward generalization of [40, Theorem 4.2]. Item (d) is Corollary 4.3.21. Item (d) is an immediate consequence of the fact that LLPO_κ is discontinuous, hence noncomputable. ☐

We now turn to the remaining reducibility and non-reducibility results which we can prove about the principles of Chapter 3. Some of these are due to Galeotti, others are due to the author. Results not due to the author are specifically advertised as such, and their proofs are omitted.

5.2 The principle $\widehat{\mathrm{LPO}}_\kappa$ and translating functions

5.2.1 Introduction

An interesting property of $\widehat{\mathrm{LPO}}$ in the classical context is that it has the same Weihrauch degree of the functions $\mathrm{EC}_\omega\colon \omega^\omega \to 2^\omega$ and $\mathrm{B}_\mathbb{R}\colon \mathbb{R}_< \to \mathbb{R}$,[1] both of which are defined in a way that is analogous to the functions EC_κ and B defined in Chapter 3.

Proposition 5.2.1. *We have* $\mathrm{B}_\mathbb{R} \equiv_\mathrm{W} \mathrm{EC}_\omega \equiv_\mathrm{W} \widehat{\mathrm{LPO}}$.

Proof. See [39, Theorem 4] for a proof of $\mathrm{B}_\mathbb{R} \equiv_\mathrm{W} \mathrm{EC}_\omega$. The fact that $\widehat{\mathrm{LPO}} \equiv_\mathrm{W} \mathrm{EC}_\omega$ is well known and it is proven as in Proposition 5.2.2. □

We point out that both EC_ω and $\mathrm{B}_\mathbb{R}$ (and their generalized counterparts) are functions serving as translations between representations of the same object. For instance, let $A \subseteq \omega$ be any set. A way of completely specifying A is by giving a sequence $p \in \omega^\omega$ such that $\mathrm{ran}(p) = A$. It is well known that this representation is "computationally poor". We give an intuition of what we mean by this. For any $n \geq 1$, let $A_n = \mathrm{ran}(p{\restriction}n)$ be the approximation of A corresponding to the information that we can obtain from the initial segment of length n of p. The only thing we can really say about the relation between the approximations A_n and A is that $A_n \subseteq A$. On the other hand, having access to an initial segment of length n of χ_A gives us an approximating set $A'_n = A \cap n$. In particular it allows us to be sure that A'_n and A differ at most on numbers greater than or equal to n.[2] Similarly, a $\delta_{\mathbb{R}_<}$-name $p \in \omega^\omega$ for $x \in \mathbb{R}$ is a computationally poor description of x as knowledge of $p{\restriction}(n+1)$ only tells us that x is greater than the rational coded by $p(n)$. This is not the case for $\delta_\mathbb{R}$-names q of x, as knowledge of $q{\restriction}(n+1)$ tells us that, letting q_n denote the rational coded by $q(n)$, $x \in [q_n - 2^{-n}, q_n + 2^{-n}]$.[3] In light of this analogy, it is rather plausible that $\mathrm{EC}_\omega \equiv_\mathrm{W} \mathrm{B}_\mathbb{R}$: a way to prove it relies on the close computational connection between subsets of the natural numbers and real numbers given by the function

$$A \mapsto s(A) = \sum_{n \in A} 2^{-(n+1)} \tag{5.1}$$

for any $A \subseteq \omega$. It is well known that this function is a bijection between the set of infinite subsets of ω and the unit interval, moreover, given any set $A \subseteq \omega$, the difference between having access to A_n and A'_n is analogous to the difference between knowing $p{\restriction}n$ for some $\delta_{\mathbb{R}_<}$-name p of $s(A)$ and knowing $q{\restriction}n$ for some $\delta_\mathbb{R}$-name q of the same real number.

In this section we present a proof of the equivalences $\widehat{\mathrm{LPO}}_\kappa \equiv_\mathrm{W} \mathrm{EC}_\kappa \equiv_\mathrm{W} \mathrm{B}$. The reduction $\mathrm{B} \leq_\mathrm{W} \mathrm{EC}_\kappa$ is obtained similarly to the corresponding reduction between their classical counterparts. The reduction $\mathrm{EC}_\kappa \leq_\mathrm{W} \mathrm{B}$ is based on a connection discovered by the author between unbounded subsets of κ and generalized reals analogous to (although not as strong as) the one established by Equation (5.1). This connection is based on Conway Normal Form (Theorem 1.4.18). The equivalence $\widehat{\mathrm{LPO}}_\kappa \equiv_\mathrm{W} \mathrm{EC}_\kappa$ follows by a straightforward computational proof analogous to the classical one.

5.2.2 The equivalence $\mathrm{EC}_\kappa \equiv_\mathrm{W} \widehat{\mathrm{LPO}}_\kappa$ and the reduction $\mathrm{B} \leq_\mathrm{W} \mathrm{EC}_\kappa$

We begin with the proof of the equivalence between EC_κ and $\widehat{\mathrm{LPO}}_\kappa$. Note that both the latter operations are single valued, total functions on the generalized Baire/Cantor spaces κ^κ and 2^κ, which we assume are represented with the identity representations. This implies that the each function is its own (only) realizer.

[1] See [7] for the formal definition of $\mathrm{B}_\mathbb{R}$ (there denoted as B) and [39] for a formal definition of EC_ω (there denoted as EC). Note that the principle $\mathrm{B}_\mathbb{R}$ appears in multiple equivalent forms (with different names) in [39] as well.

[2] Another way to see this is that a Turing machine with oracle access to $\chi_{\mathrm{ran}(p)}$ can decide $\mathrm{ran}(p)$, whereas a Turing machine with access to p can in general only semidecide $\mathrm{ran}(p)$.

[3] Note that q_n is an approximation of x accurate to a precision of 2^{-n} because the fast convergence Cauchy condition for classical computable metric spaces traditionally requires a rate of convergence given by $n \mapsto 2^{-n}$. Of course, the initial segment $q{\restriction}(n+1)$ tells us even more, namely (in the terminology of Chapter 2), that $x \in \mathrm{cl}(\mathrm{compat}_{q{\restriction}n})$.

Proposition 5.2.2. *We have* $\mathrm{EC}_\kappa \equiv_{\mathrm{sW}} \widehat{\mathrm{LPO}}_\kappa$.

Proof. We prove $\mathrm{EC}_\kappa \leq_{\mathrm{sW}} \widehat{\mathrm{LPO}}_\kappa$: define the Type 2 computable function $F\colon \kappa^\kappa \to \kappa^\kappa$ given by:

$$F(\ulcorner \alpha, \beta \urcorner) = \begin{cases} 0 & \text{if } \alpha \in \operatorname{ran}(p \upharpoonright \beta), \\ 1 & \text{otherwise.} \end{cases}$$

It is immediate to see that, for every $\alpha \in \kappa$:

$$\widehat{\mathrm{LPO}}_\kappa(F(p))(\alpha) = 0 \iff \exists \beta((F(p))_\alpha(\beta) = 0) \iff \alpha \in \operatorname{ran}(p) \iff \chi_{\operatorname{ran}(p)}(\alpha) = \mathrm{EC}_\kappa(p)(\alpha) = 1.$$

Therefore, if we let G be a simple negating function, we obtain that $G \circ \widehat{\mathrm{LPO}}_\kappa \circ F = \mathrm{EC}_\kappa$, hence $\mathrm{EC}_\kappa \leq_{\mathrm{sW}} \widehat{\mathrm{LPO}}_\kappa$.
To see that $\widehat{\mathrm{LPO}}_\kappa \leq_{\mathrm{sW}} \mathrm{EC}_\kappa$, we define the Type 2 computable function $K\colon \kappa^\kappa \to \kappa^\kappa$ given by:

$$K(\ulcorner \alpha, \beta \urcorner) = \begin{cases} \alpha + 1 & \text{if } p_\alpha(\beta) = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Again it is immediate to see that, for every $\alpha \in \kappa$,

$$\mathrm{EC}_\kappa(K(p))(\alpha+1) = 1 \iff \alpha + 1 \in \operatorname{ran}(K(p)) \iff \exists \beta((p)_\alpha(\beta) = 0) \iff \widehat{\mathrm{LPO}}_\kappa(p)(\alpha) = 0.$$

Therefore, if we define the Type 2 computable function $H\colon 2^\kappa \to \kappa^\kappa$ as

$$H(q)(\alpha) = 1 \dot{-} q(\alpha + 1),$$

we obtain that $H \circ \mathrm{EC}_\kappa \circ K = \widehat{\mathrm{LPO}}_\kappa$. This proves that $\widehat{\mathrm{LPO}}_\kappa \leq_{\mathrm{sW}} \mathrm{EC}_\kappa$. $\square$

We now prove the easier direction of the equivalence $\mathrm{EC}_\kappa \equiv_{\mathrm{W}} \mathrm{B}$.

Proposition 5.2.3. *We have* $\mathrm{B} \leq_{\mathrm{W}} \mathrm{EC}_\kappa$.

Proof. We define a function $F\colon \kappa^\kappa \to \kappa^\kappa$ as

$$F(p)(\ulcorner \alpha, \beta, \gamma \urcorner) = \begin{cases} \ulcorner \beta, \gamma \urcorner + 1 & \text{if } \alpha > \gamma \wedge |\delta_{\mathbb{Q}_\kappa}(p_\alpha) - \delta_{\mathbb{Q}_\kappa}(p_\gamma)| \geq \frac{1}{\beta+1}, \\ 0 & \text{otherwise.} \end{cases}$$

The fact that F is Type 2 computable follows from the fact that the $\mathbb{R}_\kappa$ operations and ordering restricted to $\mathbb{Q}_\kappa$ are Type 1 computable with respect to the representation $\delta_{\mathbb{Q}_\kappa}$ and from the fact that, given any ordinal β, we can compute a $\delta_{\mathbb{Q}_\kappa}$-code for $\frac{1}{\beta+1}$. We obtain that for every $p \in \kappa^\kappa$,

$$\operatorname{ran}(F(p)) = \left\{ \ulcorner \beta, \gamma \urcorner + 1 \mid \exists \alpha > \gamma \left(|\delta_{\mathbb{Q}_\kappa}(p_\alpha) - \delta_{\mathbb{Q}_\kappa}(p_\gamma)| \geq \tfrac{1}{\beta+1} \right) \right\}.$$

Therefore we know that

$$\chi_{\operatorname{ran}(F(p))}(\ulcorner \beta, \gamma \urcorner + 1) = 0 \iff \forall \alpha > \gamma \left(|\delta_{\mathbb{Q}_\kappa}(p_\alpha) - \delta_{\mathbb{Q}_\kappa}(p_\gamma)| < \tfrac{1}{\beta+1} \right), \tag{5.2}$$

for all $\alpha > \gamma$. We also define the Type 2 computable function $H :\subseteq \kappa^\kappa \to \kappa^\kappa$ as

$$H(\langle p, q \rangle)(\beta) = \begin{cases} \uparrow & \text{if } H_{\beta,q} = \{\gamma \geq (\sup(\operatorname{ran}(H(\langle p, q \rangle)) \upharpoonright \beta) + 1) \mid q(\ulcorner \beta, \gamma \urcorner + 1) = 0\} = \emptyset, \\ p(\delta) & \text{if } H_{\beta,q} \neq \emptyset \wedge \delta = \min H_{\beta,q}. \end{cases}$$

Now let $p \in \operatorname{dom}(\delta_{(\mathbb{R}_\kappa)_<})$, so p codes a strictly increasing sequence of κ-rationals converging to some κ-real x. Consider the sets of the form $H_{\beta, \chi_{\operatorname{ran}(F(p))}}$ given by

$$H_{\beta,q} = \{\gamma \in \kappa \mid \chi_{\operatorname{ran}(F(p))}(\ulcorner \beta, \gamma \urcorner + 1) = 0\}.$$

By the equivalence in (5.2) we know that each H_β is nonempty if and only if there exists $\eta \in \kappa$ such that $|\delta_{\mathbb{Q}_\kappa}(p_\alpha) - \delta_{\mathbb{Q}_\kappa}(p_\eta)| < \frac{1}{\beta+1}$ for all $\alpha > \eta$. Requiring this for every β is equivalent to requiring that the sequence coded by p is Cauchy, and this is always the case because the sequence coded by

p converges. This implies that $\{\langle p, \chi_{\mathrm{ran}(F(p))}\rangle \mid p \in \mathrm{dom}(\delta_{(\mathbb{R}_\kappa)_<})\} \subseteq \mathrm{dom}(H)$. We show that for all $p \in \mathrm{dom}(\delta_{(\mathbb{R}_\kappa)_<})$, if $\delta_{(\mathbb{R}_\kappa)_<} = x$, then $\delta_{\mathbb{R}_\kappa}(H(\langle p, \chi_{\mathrm{ran}(F(p))}\rangle)) = x$. For ease of notation, we will denote $H(\langle p, \chi_{\mathrm{ran}(F(p))}\rangle)$ as p' from now on.

First, notice that it is clear from the definition of H that the sequence p' is a subsequence of p, hence the sequence of κ-rationals coded by p' converges to x. This means that we only need to prove that p' codes a fast convergent sequence, i.e., that for all $\beta \in \kappa$ and all $\gamma > \beta$,

$$|\delta_{\mathbb{Q}_\kappa}(p'_\beta) - \delta_{\mathbb{Q}_\kappa}(p'_\gamma)| < \tfrac{1}{\beta+1}.$$

By definition of H, we have that if $p'_\beta = p_\delta$, then

$$|\delta_{\mathbb{Q}_\kappa}(p'_\beta) - \delta_{\mathbb{Q}_\kappa}(p_\gamma)| < \tfrac{1}{\beta+1}$$

holds for every $\gamma \geq \delta$, which implies that

$$|\delta_{\mathbb{Q}_\kappa}(p'_\beta) - \delta_{\mathbb{Q}_\kappa}(p'_\gamma)| < \tfrac{1}{\beta+1}$$

holds for all $\gamma > \beta$ because, again by definition of H, $p'_\gamma = p_\xi$ for some $\xi > \delta$. This shows that p' is a $\delta_{\mathbb{R}_\kappa}$-name for $\delta_{(\mathbb{R}_\kappa)_<}(p)$, theefore we have that the function,

$$p \mapsto H\langle p, \chi_{\mathrm{ran}(F(p))}\rangle = H\langle p, \mathrm{EC}_\kappa(F(p))\rangle$$

with domain given by $\in \mathrm{dom}(\delta_{(\mathbb{R}_\kappa)_<})$ is a realizer of B. This shows the required reduction. $\square$

5.2.3 Computable Conway Normal Form and the reduction $\mathrm{EC}_\kappa \leq_{\mathrm{W}} \mathrm{B}$

We now set out to proving the reduction in the other direction. As mentioned before, we first need to develop some results on transfinite sums and the Conway Normal Form for selected elements of $\mathbb{R}_\kappa$. The intuition behind our constructions is that we want to obtain a way to encode arbitrary subsets of κ as elements in $\mathbb{R}_\kappa$ in a way that is "computably invertible". We will develop the necessary analytic and computational results simultaneously.
Similarly to the classical situation, our encoding will actually only work with unbounded subsets of κ, but this is an unproblematic detail.

Definition 5.2.4 (Unbounded counterpart)**.** *Let A be any subset of κ, define its* unbounded counterpart *as*

$$U(A) = \{\beta \mid \exists \alpha \in A \ (\beta \textit{ is the } \alpha\textit{-th even ordinal })\} \cup \{\beta \in \kappa \mid \beta \textit{ odd }\}.$$

We immediately obtain:

Lemma 5.2.5. *The function $U \colon \mathcal{P}(\kappa) \to \mathcal{P}(\kappa)$ is an injection. Moreover, for all $A \subseteq \kappa$, $U(A)$ is unbounded in κ.*

Lemma 5.2.6. *The function $U_\chi \colon 2^\kappa \to 2^\kappa$ defined as $U_\chi(x) = y$ iff $y = \chi_{U(A)}$ where $\chi_A = x$ is Type 2 computable and has a (partial) Type 2 computable left inverse U_χ^{-1}, where* $\mathrm{dom}(U_\chi^{-1}) = \mathrm{ran}(U_\chi)$.

Proof. A machine computing U_χ can just print 1 on odd indexed cells and print i on the cell indexed by the α-th even ordinal, if the α-th bit of input is i. Note that this machine computes U_χ by means of an approximating function $u \colon 2^{<\kappa} \to 2^{<\kappa}$ such that for every $w \in 2^{<\kappa}$ and every $\alpha \in \mathrm{dom}(w)$, the α-th even ordinal belongs to $\mathrm{dom}(u(w))$. Computing U_χ^{-1} is simply achieved by copying the portion of input tape indexed by even ordinals on the output tape. $\square$

We now show that computing transfinite sums of surreal numbers of the form

$$\sum_{\alpha \in \kappa} \omega^{a_\alpha} \cdot r_\alpha,$$

where $(a_\alpha)_{\alpha \in \kappa}$ is a strictly decreasing cofinal sequence of κ-rationals, is a computable operation. The proof of this fact consists of several steps. We mention here that, by the results in Section 1.4.3, we know that the representations $\delta_{\mathbb{Q}_\kappa}$ and $\delta^{\mathrm{c}}_{\mathbb{Q}_\kappa}$ are computably equivalent, as well as being equivalent to the notation $\nu_{\mathbb{Q}_\kappa}$ under Hypothesis 1.2.19. Moreover, translations between different names for a κ-rational are Type 1 computable procedures. We will use these facts extensively in the rest of this section.

Lemma 5.2.7. *The function* $\exp\colon \mathbb{Q}_\kappa \to \mathbb{Q}_\kappa$ *given by the restriction of the surreal* ω*-map*

$$\exp(x) = \omega^x$$

to the κ*-rationals is* $\delta^{\mathrm{c}}_{\mathbb{Q}_\kappa}$*-computable.*

Proof. First, notice that the function exp is well defined: in view of Theorem 1.4.23, we know that the ω-map, restricted to $\mathbb{Q}_\kappa$, has range contained in $\mathbb{Q}_\kappa$.
We now describe a recursive procedure which determines a computable realizer for the function exp. On input $q \in \mathrm{dom}(\exp \circ \delta^{\mathrm{c}}_{\mathbb{Q}_\kappa})$ coding the κ-rational x, we call our machine recursively on all left and right elements of the cut for x coded by q to compute a $\delta^{\mathrm{c}}_{\mathbb{Q}_\kappa}$-name for each of the elements of

$$L = \{0, n \cdot \omega^{x_L} \mid x_L \in L_x, n \in \omega\},$$

$$R = \{1/n \cdot \omega^{x_R} \mid x_L \in R_x, n \in \omega\}.$$

By Lemma 1.4.12 we know that $[L|R] = \omega^x$. To see why these sets can be computed, recall that, as we mentioned in Section 1.4, we can always assume that $\delta^{\mathrm{c}}_{\mathbb{Q}_\kappa}$ codes represent κ-rationals as canonical cuts. Therefore, for any κ-rational, the recursively called function is completely computed in fewer than κ steps. □

We remark that the computations of the realizer of exp, as is the case for any other computable realizer of functions on the κ-rationals, can be considered as Type 1 computations by the nature of the representation $\delta_{\mathbb{Q}_\kappa}$. In a way, the previous lemma serves as the inductive step for the construction in the following:

Proposition 5.2.8. *Assume Hypothesis 1.2.19. The operation* $S :\subseteq (\mathbb{Q}_\kappa)^\kappa \times (\mathbb{Q}_\kappa)^\kappa \to (\mathbb{Q}_\kappa)^\kappa$ *given by*

$$S(e,r)(\beta) = \sum_{\alpha\in\beta} \omega^{e(\alpha)} \cdot r(\alpha),$$

with

$$\mathrm{dom}(S) = \{(e,r) \mid e \in (\mathbb{Q}_\kappa)^\kappa \downarrow \wedge\, r \in \mathbb{R}^\kappa\},$$

where we use the notation $(\mathbb{Q}_\kappa)^\kappa \downarrow$ *to denote the set of strictly decreasing sequences of* κ*-rationals, is* $[\delta_{(\mathbb{Q}_\kappa)^\kappa} \times \delta_{(\mathbb{Q}_\kappa)^\kappa}, \delta_{(\mathbb{Q}_\kappa)^\kappa}]$*-computable. Here, the representation* $\delta_{(\mathbb{Q}_\kappa)^\kappa}$ *of* $(\mathbb{Q}_\kappa)^\kappa$ *is the sequence space representation associated to the space* $(\mathbb{Q}_\kappa, \nu_{\mathbb{Q}_\kappa})$ *(see Definition 1.3.2).*

Proof. Again we first prove that the function S is well defined, i.e., that $S(e,r)(\beta) \in \mathbb{Q}_\kappa$ for all $\beta \in \kappa$ and $(e,r) \in \mathrm{dom}(S)$. To do so, fix a pair of sequences (e,r) in $\mathrm{dom}(S)$: we prove by induction on the ordinals that the surreal numbers $S(e,r)(\beta)$ are actually κ-rationals for every $\beta \in \kappa$. The claim for $\beta = 0$ is obvious. If $\beta = \gamma + 1$ and $S(e,r)(\gamma) \in \mathbb{Q}_\kappa$, notice that $\omega^{e(\gamma)} \in \mathbb{Q}_\kappa$ by Lemma 5.2.7 and $\mathbb{R} \subseteq \mathrm{No}_{\leq\omega} \subseteq \mathbb{Q}_\kappa$, so

$$S(e,r)(\beta) = S(e,r)(\gamma) + \omega^{e(\beta)} \cdot r(\beta) \in \mathbb{Q}_\kappa,$$

because $\mathbb{Q}_\kappa$ is a field. Lastly if β is a limit ordinal with $S(e,r)(\gamma) \in \mathbb{Q}_\kappa$ for all $\gamma < \beta$, then the cut (L_β, R_β) is given by a pair of sets of κ-rationals containing fewer than κ elements each. Since $\mathbb{Q}_\kappa$ is an η_κ-set, the surreal $[L_\beta|R_\beta]$ identified by it is a κ-rational.

Now we turn to the description of a computable realizer of S. Let p code a pair (e,r) in $\mathrm{dom}(S)$, we build a code q for $S(e,r)$ recursively. We have $S(e,r)(0) = 0$, therefore we can set $q(0)$ to be any $\nu_{\mathbb{Q}_\kappa}$ code for 0. Assume we have computed $q{\restriction}\beta$ and we want to compute $q(\beta)$. We make a case distinction, if $\beta = \gamma + 1$ is a successor ordinal, then

$$S(e,r)(\beta) = S(e,r)(\gamma) + \omega^{e(\beta)} \cdot r(\beta).$$

By Lemma 5.2.7 (keeping in mind that we can translate back and forth between $\nu_{\mathbb{Q}_\kappa}$ and $\delta_{\mathbb{Q}_\kappa}$ codes for rationals) we can compute a code η for $\omega^{e(\beta)}$ from the code of $e(\beta)$. Subsequently, we use computable realizers of the field operations with the code η together with the code $q(\gamma)$ and the code for $r(\beta)$ to compute a code $q(\beta)$ for $S(e,r)(\beta)$.

If β is a limit ordinal, we convert the ordinal codes in $q{\restriction}\beta$ to $\delta^{\mathrm{c}}_{\mathbb{Q}_\kappa}$ names for the rationals

$$(S(e,r)(\gamma))_{\gamma\in\beta}$$

and we use the computability of the field operations to compute a $\delta^{\mathrm{c}}_{\mathbb{Q}_\kappa}$-name for the cut $[L_\beta|R_\beta]$ where

$$L_\beta = \left\{\sum_{\alpha\in\gamma}\omega^{e(\alpha)}\cdot s(\alpha)\right\}$$

where γ ranges over β, $s(\alpha) = r(\alpha)$ for all $\alpha \in \gamma$ and s_γ ranges over $\{r(\gamma) - 1/n \mid n \in \omega\}$ and

$$R_\beta = \left\{\sum_{\alpha\in\gamma}\omega^{e(\alpha)}\cdot s'(\alpha)\right\}$$

where γ ranges over β, $s'(\alpha) = r(\alpha)$ for all $\alpha \in \gamma$ and $s'(\gamma)$ ranges over $\{r(\gamma) + 1/n \mid n \in \omega\}$. It is obvious that for every real coefficient $r(\beta)$, we can actually compute the sets $\{r(\gamma) - 1/n \mid n \in \omega\}$ and$\{r(\gamma) + 1/n \mid n \in \omega\}$. In light of this, we see that computing a $\delta^{\mathrm{c}}_{\mathbb{Q}_\kappa}$ code for $[L_\beta \mid R_\beta]$ is feasible as we can Type 1 compute $\delta^{\mathrm{c}}_{\mathbb{Q}_\kappa}$ codes for each element of either side in (L_β, R_β), and each side contains fewer than κ elements. As we mentioned in Section 1.4, the cut $[L_\beta \mid R_\beta]$ corresponds to the sum

$$S(e,r)(\beta) = \sum_{\alpha\in\beta}\omega^{e(\alpha)}\cdot r(\alpha).$$

Once we have a $\delta^{\mathrm{c}}_{\mathbb{Q}_\kappa}$-name for $S(e,r)(\beta)$, we convert it to an ordinal η such that $\nu_{\mathbb{Q}_\kappa}(\eta) = S(e,r)(\beta)$ and we set $q(\beta) = \eta$. This shows that the operation S admits a computable realizer. □

We remark that the assumption of Hypothesis 1.2.19 in the proof above is completely inessential and it is a byproduct of our choice of representations for the set $(\mathbb{Q}_\kappa)^\kappa$. The algorithm presented in the proof would work just as well if we were working with the sequence space representation of $(\mathbb{Q}_\kappa)^\kappa$ obtained from $\delta_{\mathbb{Q}_\kappa}$, and in those terms it would not need the assumption of $\mathrm{V}_\kappa = \mathrm{L}_\kappa$. The reason the proposition is stated in terms of $\nu_{\mathbb{Q}_\kappa}$ is to be uniform with its end-goal, Proposition 5.2.15, which crucially needs the assumption of Hypothesis 1.2.19.

We turn to some analytical properties of the function S. First we show that for every (e,r) such that e is coinitial in $\mathbb{Q}_\kappa$, the sequence $(s_\beta)_{\beta\in\kappa} = S(e,r)(\beta)$ is Cauchy, hence convergent in $\mathbb{R}_\kappa$. Further, for every $(e,r) \in \mathrm{dom}(S)$, the limit of the sequence $(S(e,r)(\beta))_{\beta\in\kappa}$ coincides with the full sum $\sum_{\alpha\in\kappa}\omega^{e(\alpha)}\cdot r(\alpha)$.

Lemma 5.2.9. *Let $(e,r) \in \mathrm{dom}(S)$ be such that e is coinitial in $\mathbb{Q}_\kappa$ and let $(s_\beta)_{\beta\in\kappa} = (S(e,r)(\beta))_{\beta\in\kappa}$. The sequence $(s_\beta)_{\beta\in\kappa}$ is Cauchy, hence convergent, in $\mathbb{R}_\kappa$.*

Proof. To prove that $(s_\beta)_{\beta\in\kappa}$ is Cauchy, we need to show that for every $r \in \mathbb{R}^+_\kappa$, there is an ordinal γ such that if $\eta, \xi > \gamma$, then

$$|s_\eta - s_\xi| < r.$$

Since the set $\{1/\alpha \mid 0 < \alpha < \kappa\}$ is coinitial in $\mathbb{R}^+_\kappa$, it is enough to show that for every $0 < \beta \in \kappa$, there is an ordinal γ such that if $\eta, \xi > \gamma$, then

$$|s_\eta - s_\xi| < \tfrac{1}{\beta}.$$

To this end, fix a nonzero ordinal $\beta < \kappa$. Since e is coinitial in $\mathbb{Q}_\kappa$, there is an index γ such that $e(\gamma) < -\beta$. Now for any $\eta, \xi > \gamma$, the tail property (equation 1.1 in Theorem 1.4.16) entails that

$$|s_\eta - s_\xi| \ll \omega^{e(\gamma)}.$$

Therefore, since $e(\gamma) < -\beta$ and $\beta \le \omega^\beta$,

$$|s_\eta - s_\xi| < \tfrac{1}{\beta}.$$

This shows that $(s_\beta)_{\beta\in\kappa}$ is Cauchy, and since $\mathbb{R}_\kappa$ is Cauchy complete, $(s_\beta)_{\beta\in\kappa}$ converges to some $s \in \mathbb{R}_\kappa$. □

Lemma 5.2.10. *Let $(e,r) \in \mathrm{dom}(S)$ be such that e is coinitial in $\mathbb{Q}_\kappa$, let $(s_\beta)_{\beta\in\kappa} = (S(e,r)(\beta))_{\beta\in\kappa}$ and let $s = \lim_{\beta\in\kappa} s_\beta$. We have that*

$$s = s_\kappa = \sum_{\alpha\in\kappa} \omega^{e(\alpha)} \cdot r(\alpha).$$

Proof. Let (e',r') be the reduced sequence associated to (e,r). There are two possible options: if $\mathrm{dom}(r') = \lambda < \kappa$, then by the regularity of κ there must be some λ' such that $r(\eta) = 0$ for every $\eta > \lambda'$. In this case, $(s_\alpha)_{\alpha\in\kappa}$ is eventually constant with value $s_{\lambda'}$, therefore it is trivially convergent. In case $\lambda = \kappa$, define the sets of κ-rationals L_κ and R_κ as

$$L_\kappa = \{s_\beta + \omega^{e(\beta)} \cdot r'(\beta) \mid r'(\beta) \in \mathbb{R}_{<r(\beta)}, \beta \in \kappa\},$$

and

$$R_\kappa = \{s_\beta + \omega^{e(\beta)} \cdot r'(\beta) \mid r'(\beta) \in \mathbb{R}_{>r(\beta)}, \beta \in \kappa\}.$$

By definition we have $s_\kappa = [L_\kappa|R_\kappa]$. We claim that $L < s < R$: suppose by contradiction that L does not lie below s, then there must be an ordinal β and a real number $\ell < r(\beta)$ such that $s \le s_\beta + \omega^{e(\beta)} \cdot \ell$, which implies in particular that

$$|s - s_{\beta+1}| \ge |s_{\beta+1} - (s_\beta + \omega^{e(\beta)} \cdot \ell)| = \omega^{e(\beta)} \cdot (r(\beta) - \ell) \sim_a \omega^{e(\beta)}. \tag{5.3}$$

The sequence $(s_\alpha)_{\alpha\in\kappa}$ converges to s. So, in particular there is an index η such that for all $\gamma \ge \eta$, we have

$$|s - s_\gamma| < \omega^{e(\beta)-1} \ll \omega^{e(\beta)}.$$

Further, by the tail property we obtain that for all $\gamma > \beta + 1$,

$$|s_\gamma - s_{\beta+1}| \ll \omega^{e(\beta)}.$$

Therefore, if we pick $\gamma' > \max\{\eta, \beta + 1\}$ we obtain that

$$|s - s_{\beta+1}| \le |s - s_{\gamma'}| + |s_{\gamma'} - s_{\beta+1}| \ll \omega^{e(\beta)},$$

since, if x, y and a are any non-negative surreal numbers such that $x \ll a$ and $y \ll a$, then it follows that $x + y \ll a$. This is in contradiction with (5.3). An analogous argument shows that $s < R$, therefore we obtain $L < s < R$. By the Fundamental Exitence Theorem (Theorem 1.4.3), we get $s_\kappa \subseteq s$. Now by Lemma 1.4.17, we know that $l(s_\kappa) \ge \kappa$ and by Theorem 1.4.26 we know that $l(s) \le \kappa$ as $s \in \mathbb{R}_\kappa$. This means that the only possibility is that $l(s) = l(s_\kappa) = \kappa$ and $s = s_\kappa$. ☐

Definition 5.2.11 (Limit map). *We define the* limit map $l :\subseteq (\mathbb{Q}_\kappa)^\kappa \to \mathbb{R}_\kappa$ *as*

$$l((q_\alpha)_{\alpha\in\kappa}) = \lim_{\alpha\in\kappa}((q_\alpha)_{\alpha\in\kappa}),$$

with $\mathrm{dom}(l) = \{(q_\alpha)_{\alpha\in\kappa} \in (\mathbb{Q}_\kappa)^\kappa \mid (q_\alpha)_{\alpha\in\kappa}$ *converges*$\}$.

By Lemma 5.2.10, we can form the composition $l \circ S'$ where S' is the restriction of the function S to pairs (e,r) where e is coinitial in $\mathbb{Q}_\kappa$.

Lemma 5.2.12 (Computable transfinite sums). *Assume Hypothesis 1.2.19. The function $T :\subseteq (\mathbb{Q}_\kappa)^\kappa \times (\mathbb{Q}_\kappa)^\kappa \to \mathbb{R}_\kappa$ defined as $T = l \circ S'$ is $(\delta_{(\mathbb{Q}_\kappa)^\kappa} \times \delta_{(\mathbb{Q}_\kappa)^\kappa}, \delta_{\mathbb{R}_\kappa})$-computable.*

Proof. By Proposition 5.2.8 there is a T2κTM M which computes a realizer of S'. Given a sequence p coding a pair $(e,r) \in \mathrm{dom}(T)$, we simulate the machine M on p. The simulated output is a $\delta_{(\mathbb{Q}_\kappa)^\kappa}$ code for the sequence $(s_\beta)_{\beta\in\kappa}$ defined as

$$s_\beta = \sum_{\alpha\in\beta} \omega^{e(\alpha)} \cdot r(\alpha).$$

By Lemma 5.2.10, this sequence converges to $y = T(e, r)$. This in particular means that $(s_\beta)_{\beta\in\kappa}$ is Cauchy. More precisely, we know that e is coinitial in $\mathbb{Q}_\kappa$, hence if we let δ' be any ordinal such that $e(\delta') < -\gamma$, the tail property immediately implies that

$$|s_\eta - s_\xi| \ll \omega^{-\gamma} \sim_a \tfrac{1}{\gamma+1},$$

for every $\eta, \xi > \delta'$. This shows that any such δ' is a witness for the fact that $(s_\beta)_{\beta\in\kappa}$ satisfies the instance of the Cauchy condition for the bound $\frac{1}{\gamma+1}$. It is clear that for any given γ, it is possible to compute the least such δ'. This implies that we can compute (uniformly in the code p) the sequence $q_p \in \kappa^\kappa$ defined as:

$$q_p(\alpha) = \min\{\delta \mid e(\delta) < -\gamma\}.$$

A computable realizer N for the function T then behaves as follows: on input p, it computes the sequence of rationals coded by $M(p)$ as well as the sequence of ordinals q_p. While doing this, it prints the subsequence $N(p)$ of $M(p)$ given by

$$N(p)(\alpha) = M(p)(q_p(\alpha)).$$

By construction $N(p)$ codes a fast convergent Cauchy sequence with limit $T(e, r)$, i.e., T is computable. $\square$

In the same way as for Proposition 5.2.8, the assumption of Hypothesis 1.2.19 is inessential. Notice that by Theorem 1.4.27 we know that $\mathrm{ran}(T) = \mathbb{R}_\kappa$.

Lemma 5.2.13. *Assume Hypothesis 1.2.19. The function T admits a $[\delta_{\mathbb{R}_\kappa}, \delta_{(\mathbb{Q}_\kappa)^\kappa} \times \delta_{(\mathbb{Q}_\kappa)^\kappa}]$-computable inverse* CNF: $\mathbb{R}_\kappa \to (\mathbb{Q}_\kappa)^\kappa \times (\mathbb{Q}_\kappa)^\kappa$.

Proof. Again we exploit the computability of the function S. Let M be a T2κTM which computes the realizer described in the proof of Proposition 5.2.8. Let $m :\subseteq \kappa^{<\kappa} \to \kappa^{<\kappa}$ be the Type 1 computable function associated to M as in Lemma 1.2.21. By inspection of the proof of Proposition 5.2.8 it is immediate to see that if $w \subseteq q$ for some $q \in \mathrm{dom}(S \circ \delta_{(\mathbb{Q}_\kappa)^\kappa \times (\mathbb{Q}_\kappa)^\kappa})$, then $|w| = |m(w)|$ and the word $m(w)$ consists of a sequence of ordinals coding (via the notation $\nu_{\mathbb{Q}_\kappa}$) the sequence of rationals $(s_\alpha)_{\alpha\in|w|}$ corresponding to the partial sums. Notice that for any given short sequence of ordinals w, we have that the machine M can Type 1 decide whether w can be extended to a code $q \in \mathrm{dom}(S \circ \delta_{(\mathbb{Q}_\kappa)^\kappa \times (\mathbb{Q}_\kappa)^\kappa})$. Call this set of "extendible words" A.

Consider the machine N which has a scratch tape T_i reserved as an ordinal register and, on input $p \in \mathrm{dom}(\delta_{\mathbb{R}_\kappa})$, in parallel, generates all words $w \in A$ with $|w|$ a successor ordinal, computes sequences $m(w)$ and parses the code p until it finds a word $w' \in \kappa^{<\kappa}$ and an ordinal β such that:

(a) the word w' codes a pair of sequences $((e_\alpha)_{\alpha\in\lambda+2}, (r_\alpha)_{\alpha\in\lambda+2})$ for some λ and, if $(s_\alpha)_{\alpha\in\lambda+2}$ is the sequence of partial sums coded by $m(w')$, we have that $|s_{\lambda+1} - \nu_{\mathbb{Q}_\kappa}(p(\beta))| < \omega^{e(\lambda+1)-1}$,

(b) it is the case that $\frac{1}{\beta+1} < \omega^{e(\lambda+1)-1} \ll \omega^{e(\lambda+1)}$,

We claim that whenever this is the case, the reduced sequence $((e'_\alpha)_{\alpha\in\gamma}, (r'_\alpha)_{\alpha\in\gamma})$ corresponding to $((e_\alpha)_{\alpha\in\lambda+1}, (r_\alpha)_{\alpha\in\lambda+1})$ is an initial segment of the sequence of exponent-coefficient pairs in the Conway Normal Form of $T(\delta_{\mathbb{R}_\kappa}(p))$. To see this, notice that

$$|\nu_{\mathbb{Q}_\kappa}(p(\beta)) - \delta_{\mathbb{R}_\kappa}(p)| \le \tfrac{1}{\beta+1} \ll \omega^{e(\lambda+1)} \text{ and } |\nu_{\mathbb{Q}_\kappa}(p(\beta)) - s_{\lambda+1}| < \omega^{e(\lambda+1)-1} \ll \omega^{e(\lambda+1)},$$

where the first inequality is due to the fast convergence condition on p together with our assumption on β and the second is by assumption on $\nu_{\mathbb{Q}_\kappa}(p(\beta))$. By the triangle inequality, we get $|s_{\lambda+1} - \delta_{\mathbb{R}_\kappa}(p)| \ll \omega^{e(\lambda+1)}$, so by Lemma 1.4.19, their Conway Normal Forms coincide up to exponents below $e(\lambda+1)$, i.e., $((e'_\alpha)_{\alpha\in\gamma}, (r'_\alpha)_{\alpha\in\gamma})$ is the beginning of the Conway Normal Form for $\delta_{\mathbb{R}_\kappa}(p)$.

Now, whenever N finds a word w' coding the sequence $((e_\alpha)_{\alpha\in\lambda+2}, (r_\alpha)_{\alpha\in\lambda+2})$ and an ordinal β satisfying the conditions above, it performs the following actions: if $r(\lambda) \neq 0$, it computes a word w'' coding the reduced sequence $((e'_\alpha)_{\alpha\in\lambda'}, (r'_\alpha)_{\alpha\in\lambda'})$ corresponding to $((e_\alpha)_{\alpha\in\lambda+1}, (r_\alpha)_{\alpha\in\lambda+1})$ (doing so just requires erasing those pairs which feature a null real coefficient), then it updates the output tape to match the word w'' in the following sense: if the output tape contains a word t coding a (potentially non-reduced) sequence $((f_\alpha)_{\alpha\in\gamma}, (s_\alpha)_{\alpha\in\gamma})$, then we know that the reduced sequence $((f'_\alpha)_{\alpha\in\gamma'}, (s'_\alpha)_{\alpha\in\gamma'})$ must be compatible with $((e'_\alpha)_{\alpha\in\lambda'}, (r'_\alpha)_{\alpha\in\lambda'})$. In case the former is shorter than the latter, N appends bits to the end of the output tape to make them coincide. The machine N then updates the tape T_i to contain the least ordinal η such that $-\eta < e(\lambda)$.

On the other hand, assume w' codes a pair of sequences $((e_\alpha)_{\alpha\in\lambda+2}, (r_\alpha)_{\alpha\in\lambda+2})$ with $r(\lambda) = 0$, then again we know that the reduced sequence $((e'_\alpha)_{\alpha\in\lambda'}, (r'_\alpha)_{\alpha\in\lambda'})$ corresponding to $((e_\alpha)_{\alpha\in\lambda+1}, (r_\alpha)_{\alpha\in\lambda+1})$ is an initial segment of the Conway Normal Form of $\delta_{\mathbb{R}_\kappa}(p)$. In this situation, the machine checks whether $e(\lambda+1) \leq -\eta$ where η is the current content of the tape T_i. If so, it computes a word w'' coding $((e'_\alpha)_{\alpha\in\lambda'}, (r'_\alpha)_{\alpha\in\lambda'})$ and concatenates it with a code for $(e(\lambda), r(\lambda))$, then updates the output tape to match this word as above. If, on the other hand $e(\lambda+1) > -\eta$, then the machine ignores the word w, β and goes on in its computation. Notice that we can exploit the lim sup rule to make sure that the content of T_i is set to the supremum of its values at any limit ordinal step of computation.

This behaviour is constructed so that the machine N appends a code for a "useless" pair $(e(\lambda), r(\lambda))$ where $r(\lambda) = 0$ only when the exponent $e(\lambda)$ is "sufficiently far" from the previous exponents. As we will show, this is sufficient to ensure that N behaves correctly.

It is clear that for any $p \in \mathrm{dom}(\delta_{\mathbb{R}_\kappa})$ and any ξ, the content of the output tape at time step ξ codes a sequence which, when reduced, is an initial segment of the Conway Normal Form of $\delta_{\mathbb{R}_\kappa}(p)$. It is also straightforward to see that the machine N runs indefinitely on all inputs $p \in \mathrm{dom}(\delta_{\mathbb{R}_\kappa})$. So, all we have to show is that on any input $p \in \mathrm{dom}(\delta_{\mathbb{R}_\kappa})$, N produces a code for the full Conway Normal Form of p.

Since the reduced counterpart of the sequence coded by the output tape at any time is an initial segment of the Conway Normal Form of $\delta_{\mathbb{R}_\kappa}(p)$, this boils down to showing that, if $N(p)$ codes the sequence $((e_\alpha)_{\alpha\in\kappa}, (r_\alpha)_{\alpha\in\kappa})$, and for some $\lambda \in \kappa$, it is the case that $r(\eta) = 0$ for all $\eta \geq \lambda$, then it must be that the reduced sequence corresponding to $((e_\alpha)_{\alpha\in\lambda}, (r_\alpha)_{\alpha\in\lambda})$ is already the Conway Normal Form of $\delta_{\mathbb{R}_\kappa}(p)$. Notice that at any step of computation, the ordinal η contained in the tape T_i is such that $-\eta$ is a lower bound for the exponents coded by the current output. Moreover, the ordinal counter T_i increases by at least 1 each time the machine N prints a code for a pair $(e, 0)$. An easy induction then shows that, at any step of computation and for any ordinal β, if the output tape codes a sequence containing β-many null real coefficients, then for every $\alpha < \beta$, the Conway Normal Form of $\delta_{\mathbb{R}_\kappa}(p)$ up to exponents $\leq -\alpha$ is coded by the current output tape. This immediately implies what we wanted to show, therefore concluding the proof that N computes a realizer of CNF. □

The last technical tool that we need to introduce is the following function, which is in practice a cleanup routine for the Conway Normal Form of κ-reals which have a κ-long Conway Normal Form.

Definition 5.2.14 (Cleanup function)**.** *We define the* cleanup function $c :\subseteq \mathbb{Q}_\kappa^\kappa \times \mathbb{Q}_\kappa^\kappa \to \mathbb{Q}_\kappa^\kappa \times \mathbb{Q}_\kappa^\kappa$*, as*

$$c(e, r) = (e', r'),$$

where $\mathrm{dom}(c) = \{(e, r) \in \mathrm{dom}(T) \mid r \textit{ is not eventually } 0\}$ *and* (e', r') *is the reduced sequence corresponding to* (e, r)*.*

It is clear that the function c has a computable realizer.

We can finally prove the reverse direction of Proposition 5.2.3.

Proposition 5.2.15. *Assuming Hypothesis 1.2.19, we have* $\mathrm{EC}_\kappa \leq_{\mathrm{sW}} \mathrm{B}$.

Proof. We want to show that there are Type 2 computable functions $H, K :\subseteq \kappa^\kappa \to \kappa^\kappa$ such that for all $p \in \kappa^\kappa$ and for all realizers A of B, we have that $H(A(K(p))) = \chi_{\mathrm{ran}(p)}$. We build a function K such that for every $p \in \kappa^\kappa$, and every A realizing B the real $\delta_{\mathbb{R}_\kappa}(K(p))$ fully encodes $\mathrm{ran}(p)$ in its Conway Normal Form.

We define the function $F\colon 2^\kappa \to (\mathbb{Q}_\kappa)^\kappa \times (\mathbb{Q}_\kappa)^\kappa$ given by

$$F(x) = ((-\alpha)_{\alpha\in\kappa}, (x(\alpha) + 1)_{\alpha\in\kappa}),$$

in other words, function F associates to any $x \in 2^\kappa$ the sequence of exponent-coefficient pairs where the exponents are the additive inverses of all the ordinals in κ and the α-th coefficient is 2 if $x(\alpha) = 1$ and 1 if $x(\alpha) = 0$. It is clear that F is computable and that it admits a computable inverse F'. Moreover, we define the function $G \colon \kappa^{<\kappa} \to 2^\kappa$ given by $G(w) = U_\chi(w')$, where $w' = \chi_{\mathrm{ran}(w)}$. We immediately see that G is also Type 2 computable. Lastly let R be the computable realizer of the function S defined in Proposition 5.2.8. Define the Type 2 computable function $K \colon \kappa^\kappa \to \kappa^\kappa$ as

$$K(p)(\beta) = R(F(G(p\restriction\beta)))(\beta).$$

An explanation is in order: what this means is that $K(p)(\beta)$ corresponds to a $\nu_{\mathbb{Q}_\kappa}$-code for the κ-rational

$$s_\beta = \sum_{\alpha\in\beta} \omega^{-\alpha} \cdot v_\alpha,$$

where $v_\alpha = 1$ if $\alpha \notin \mathrm{ran}(U_\chi(p\restriction\beta))$ and where $v_\alpha = 2$ if $\alpha \in \mathrm{ran}(U_\chi(p\restriction\beta))$. In other words, the κ-rational s_β has Conway Normal Form given by $F(G(p\restriction\beta))\restriction\beta$.

We claim that, for all $p \in \kappa^\kappa$, the sequence $K(p)$ codes a strictly increasing sequence of κ-rationals: if $\alpha < \gamma$, then $\mathrm{ran}(p\restriction\alpha) \subseteq \mathrm{ran}(p\restriction\gamma)$, hence $U_\chi((p\restriction\alpha))(\eta) \geq U_\chi((p\restriction\gamma))(\eta)$ for all η. Therefore we get $F(G(p\restriction\alpha)) \leq F(G(p\restriction\gamma))$ in the lexicographical order the sequences of exponent-coefficient pairs (cf. Section 1.4), and in particular, since all the coefficients are positive $F(G(p\restriction\alpha))\restriction\alpha < F(G(p\restriction\gamma))\restriction\gamma$ in the same ordering. The fact that the surreal ordering coincides with the ordering of sequences on their Conway Normal Forms (cf. Theorem 1.4.16) then establishes that $\delta_{(\mathbb{Q}_\kappa)^\kappa}(K(p))$ is strictly increasing.

Now notice that G is not monotone, but it has the property that for all $p \in \kappa^\kappa$, the sequence $(G(p\restriction\alpha))_{\alpha\in\kappa}$ is pointwise convergent and its pointwise limit is the function $U_\chi(\chi_{\mathrm{ran}(p)}) = U_\chi(\mathrm{EC}_\kappa(p))$.

By construction it is clear that, for all $p \in \kappa^\kappa$, the sequences $\delta_{(\mathbb{Q}_\kappa)^\kappa}(K(p))$ and $S(F(U_\chi(\mathrm{EC}_\kappa(p))))$ are mutually cofinal, and by Proposition 5.2.8 $S(F(U_\chi(\mathrm{EC}_\kappa(p))))$ converges to some real s_p, hence $\delta_{(\mathbb{Q}_\kappa)^\kappa}(K(p))$ converges to s_p too. Notice that by Lemma 5.2.10, the sequence $F(U_\chi(\mathrm{EC}_\kappa(p)))$ is the (reduced) Conway Normal form of s_p. Now, by definition of $\delta_{(\mathbb{R}_\kappa)_<}$ we see that $K(p)$ is a $\delta_{(\mathbb{R}_\kappa)_<}$-name for s_p. Hence, for any realizer A of B, $K(p) \in \mathrm{dom}(A)$ and $A(K(p))$ is a $\delta_{\mathbb{R}_\kappa}$-name for s_p.

Now let $D :\subseteq \kappa^\kappa \to \kappa^\kappa$ be a computable realizer of the function $c \circ \mathrm{CNF}$ and let E be a computable realizer of F^{-1} and define $H = U_\chi^{-1} \circ E \circ D$: we get that, for all $p \in \kappa^\kappa$

$$H(A(K(p))) = U_\chi^{-1} \circ E \circ D(A(K(p))) = U_\chi^{-1} \circ E \circ D(s_p) = U_\chi^{-1}(U_\chi(\mathrm{EC}_\kappa(p))) = \mathrm{EC}_\kappa(p),$$

where the second to last equality is due to the fact that $D(s_p)$ is a code for the (reduced) Conway Normal Form of s_p, hence $E(D(s_p))$ is the unique $x \in 2^\kappa$ such that $F(x) = c \circ \mathrm{CNF}(s_p)$ i.e., $x = U_\chi(\mathrm{EC}_\kappa(p))$. ☐

Corollary 5.2.16. *Assuming Hypothesis 1.2.19, we have* $\mathrm{B} \equiv_W \mathrm{EC}_\kappa \equiv_\mathrm{W} \widehat{\mathrm{LPO}}_\kappa$.

While we were developing Proposition 5.2.15, we produced some results on a way of seeing $\mathbb{R}_\kappa$ in terms of transfinite sums as side effects. We summarize them here.

First we can define yet another representation of $\mathbb{R}_\kappa$.

Definition 5.2.17 (Sum representation of $\mathbb{R}_\kappa$)**.** *Define the representation* $\delta^{\mathrm{s}}_{\mathbb{R}_\kappa} :\subseteq \kappa^\kappa \to \mathbb{R}_\kappa$ *as*

$$\delta^{\mathrm{s}}_{\mathbb{R}_\kappa}(p) = r,$$

if and only if $T(\delta_{(\mathbb{Q}_\kappa)^\kappa \times (\mathbb{Q}_\kappa)^\kappa})(p) = r$.

Lemma 5.2.18. *Assuming Hypothesis 1.2.19, we have* $\delta^{\mathrm{s}}_{\mathbb{R}_\kappa} \equiv_\mathrm{c} \delta_{\mathbb{R}_\kappa}$.

Proof. We neeed to show that there are Type 2 computable functions H and K such that for all $r \in \mathbb{R}_\kappa$ and all $\delta^{\mathrm{s}}_{\mathbb{R}_\kappa}$-codes p for r, $H(p)$ is a $\delta_{\mathbb{R}_\kappa}$-code for r and viceversa, for any $\delta_{\mathbb{R}_\kappa}$-code q for r, $K(q)$ is a $\delta^{\mathrm{s}}_{\mathbb{R}_\kappa}$-code for r. By Lemma 5.2.12, any computable realizer of T works as H, while by Lemma 5.2.13, a computable realizer of CNF works as K. ☐

We mention here that in the literature one can find algorithms to compute the surreal field operations in terms of sum representations, see e.g., [21], Theorem 5.7 and Theorem 5.8. We conjecture that adaptations of these can be used to provide alternative proofs of the computability of the $\mathbb{R}_\kappa$ operations.

The sum representation also affords us a relatively natural, computable correspondence between special subsets of $\mathbb{Q}_\kappa$ and elements of $\mathbb{R}_\kappa$. As $-\kappa \subseteq \mathbb{Q}_\kappa$, this in turn yields a computable correspondence between unbounded subsets of κ and κ-reals reminiscent of the correspondence in the classical setting.

5.3 The principles $\widehat{\mathrm{LLPO}}_\kappa$, Sep_κ, and trees

The parallelized lesser principle of omniscience $\widehat{\mathrm{LLPO}}$ is central in the classification of classical choice and boundedness principle of [7]. In this section we present what we currently know about the status of $\widehat{\mathrm{LLPO}}_\kappa$ and the generalizations of the principles which are classically related to it.

We start by proving that, as in the classical setting, $\widehat{\mathrm{LLPO}}_\kappa = \mathrm{Sep}_\kappa$. The proof is a rather straightforward computation, which can be carried out in the same exact way both in the generalized setting and in the classical setting.

Proposition 5.3.1. *We have* $\widehat{\mathrm{LLPO}}_\kappa \equiv_{\mathrm{sW}} \mathrm{Sep}_\kappa$.

Proof. We first show that $\mathrm{Sep}_\kappa \leq_{\mathrm{sW}} \widehat{\mathrm{LLPO}}_\kappa$: define the Type 2 computable function $K :\subseteq \kappa^\kappa \to \kappa^\kappa$ as

$$K(\langle p,q\rangle)(\ulcorner \alpha,\beta \urcorner) = \begin{cases} 1 & \text{if } \beta \text{ is the } \gamma\text{-th odd ordinal for } \gamma = \min\{\eta \mid p(\eta)=\alpha\}, \\ 1 & \text{if } \beta \text{ is the } \gamma\text{-th even ordinal for } \gamma = \min\{\eta \mid q(\eta)=\alpha\}, \\ 0 & \text{otherwise.} \end{cases}$$

with $\mathrm{dom}(K) = \mathrm{dom}(\mathrm{Sep}_\kappa)$. We obtain that $(K(\langle p,q\rangle))_\alpha$ has a non-zero entry on an in odd indexed position if and only if $\alpha \in \mathrm{ran}(p)$, similarly it has a non-zero entry in an even indexed position if and only if $\alpha \in \mathrm{ran}(q)$. For $\langle p,q\rangle \in \mathrm{dom}(\mathrm{Sep}_\kappa)$ we know that $\mathrm{ran}(p) \cap \mathrm{ran}(q) \neq \emptyset$, therefore $K(\langle p,q\rangle) \in \mathrm{dom}(\widehat{\mathrm{LLPO}})_\kappa$. Now, let L be any realizer of $\widehat{\mathrm{LLPO}}_\kappa$, $\langle p,q\rangle \in \mathrm{dom}(\mathrm{Sep}_\kappa)$ and $\alpha \in \kappa$: we obtain

$$L(K(\langle p,q\rangle))(\alpha) = 0 \implies \forall \alpha \text{ even } K(\langle p,q\rangle)(\alpha) = 0 \iff \alpha \notin \mathrm{ran}(q),$$

and similarly

$$L(K(\langle p,q\rangle))(\alpha) = 1 \implies \forall \alpha \text{ odd } K(\langle p,q\rangle)(\alpha) = 0 \iff \alpha \notin \mathrm{ran}(p).$$

Now let $H \colon 2^\kappa \to 2^\kappa$ be the negating function

$$H(p)(\alpha) = 1 - p(\alpha),$$

then, for every $\langle p,q\rangle \in \mathrm{dom}(\mathrm{Sep}_\kappa)$ and every realizer L of $\widehat{\mathrm{LLPO}}_\kappa$:

$$H(L(K(\langle p,q\rangle))) = 1 \iff \alpha \notin \mathrm{ran}(q).$$

Therefore $H(L(K(\langle p,q\rangle)))$ is the characteristic function of the set $\kappa \setminus \mathrm{ran}(q)$, i.e., $H(L(K(\langle p,q\rangle))) \in \mathrm{Sep}_\kappa(\langle p,q\rangle)$. This shows that $H \circ L \circ K$ is a realizer of Sep_κ, hence $\mathrm{Sep}_\kappa \leq_{\mathrm{sW}} \widehat{\mathrm{LLPO}}_\kappa$.

Conversely, we define the Type 2 computable function $G :\subseteq \kappa^\kappa \to \kappa^\kappa$ as, for every $p \in \mathrm{dom}(\widehat{\mathrm{LLPO}}_\kappa)$,

$$G(p) = \langle \ell_p, r_p \rangle,$$

where

$$\ell_p(\gamma) = \begin{cases} \alpha+1 & \text{if } \eta \text{ is the } \gamma\text{-th even ordinal and } p_\alpha(\eta) = 1, \\ 0 & \text{otherwise.} \end{cases}$$

and r_p is defined similarly for odd ordinals. We obtain, for all $\alpha \in \kappa$ and $p \in \mathrm{dom}(\widehat{\mathrm{LLPO}}_\kappa)$,

$$\alpha + 1 \in \mathrm{ran}(r_p) \iff 1 \notin \mathrm{LLPO}_\kappa(p_\alpha) \implies 0 \in \mathrm{LLPO}_\kappa(p_\alpha).$$

and similarly

$$\alpha+1 \in \operatorname{ran}(\ell_p) \iff 0 \notin \mathrm{LLPO}_\kappa(p_\alpha) \implies 1 \in \mathrm{LLPO}_\kappa(p_\alpha).$$

Now let S be any realizer for Sep_κ, we know that $S(G(p)) = \chi_A$ where A is some subset of κ such that $\operatorname{ran}(\ell_p) \subseteq A$, $\operatorname{ran}(q) \cap A = \emptyset$. We obtain, for all $\alpha \in \kappa$:

$$S(G(p))(\alpha+1) = 1 \implies \alpha+1 \notin \operatorname{ran}(r_p) \implies 1 \in \mathrm{LLPO}_\kappa(p_\alpha),$$

$$S(G(p))(\alpha+1) = 0 \implies \alpha+1 \notin \operatorname{ran}(\ell_p) \implies 0 \in \mathrm{LLPO}_\kappa(p_\alpha).$$

So we can define the Type 2 computable function $F\colon 2^\kappa \to 2^\kappa$ as

$$F(x)(\alpha) = 1 - x(\alpha+1),$$

and obtain that $F \circ S \circ G(p) \in \widehat{\mathrm{LLPO}}_\kappa(p)$ for all $p \in \operatorname{dom}(\widehat{\mathrm{LLPO}}_\kappa)$, i.e $F \circ S \circ G$ is a realizer of $\widehat{\mathrm{LLPO}}_\kappa$. This shows that $\widehat{\mathrm{LLPO}}_\kappa \leq_{\mathrm{sW}} \mathrm{Sep}_\kappa$ and thus $\mathrm{Sep}_\kappa \equiv_{\mathrm{sW}} \widehat{\mathrm{LLPO}}_\kappa$. □

We introduce a weakening of the principle WKL_κ:

Definition 5.3.2 (Restriction of weak Kőnig's Lemma)**.** *Define the function* RWKL_κ *as the restriction of* WKL_κ *to the set of trees with* $<\kappa$*-closed extendible parts, i.e.*

$$\operatorname{dom}(\mathrm{RWKL}_\kappa) = \{T \in \mathrm{Tr} \mid \operatorname{ext}(T) \text{ is } <\kappa\text{-closed}\}.$$

We will show that this function has the same topological Weihrauch degree of the principles Sep_κ and $\widehat{\mathrm{LLPO}}_\kappa$. This is in analogy with the classical equivalences $\mathrm{WKL} \equiv_{\mathrm{W}} \mathrm{Sep} \equiv_{\mathrm{W}} \widehat{\mathrm{LLPO}}$. Moreover, it is in line with the analogy between trees of height ω and trees of height κ with $<\kappa$-closed extendible parts we obtain by comparing Example 1.1.36 and Lemma 1.1.38. We remark that the following results employ Hypothesis 1.2.19. This seems to be necessary when working with trees as working with our representation of the set of trees, which is arguably the only reasonable representation, requires being able to computably go back and forth between ordinals in κ and the sequences in $2^{<\kappa}$ that they code.

Lemma 5.3.3. *Assuming Hypothesis 1.2.19, we have* $\mathrm{Sep}_\kappa \leq_{\mathrm{sW}} \mathrm{RWKL}_\kappa$.

Proof. We define the Type 2 computable function $f\colon \kappa^\kappa \times \kappa \to \kappa^\kappa$ as

$$f(p,\alpha) = (p{\restriction}(\alpha+1))^\frown p(0).$$

The function f is needed as a technical tool to simplify the definition of the function K below, which translates pairs of sequences into trees. We define the function $K\colon \kappa^\kappa \to \mathrm{Tr}$ as

$$K(\langle p,q\rangle) = \{\sigma \in 2^{<\kappa} \mid \exists x \in \mathrm{Sep}_\kappa(\langle f(p,|\sigma|), f(q,|\sigma|)\rangle)\,(\sigma \subseteq x)\}.$$

In words, a short sequence $\sigma \in 2^{<\kappa}$ is enumerated in $K(\langle p,q\rangle)$ if and only if it is the characteristic function of a set $A_\sigma \subseteq |\sigma|$ separating $\operatorname{ran}(p{\restriction}|\sigma|)$ and $\operatorname{ran}(q{\restriction}|\sigma|)$. It is clear that K admits a Type 2 computable realizer.

We claim that for every $\langle p,q\rangle \in \operatorname{dom}(\mathrm{Sep}_\kappa)$, $K(\langle p,q\rangle)$ is a tree with at least a branch and a $<\kappa$-closed extendible part.

The fact that $K(\langle p,q\rangle)$ is a tree is trivial from the definition, moreover, it is immediate to see that if A is any set separating $\operatorname{ran}(p)$ and $\operatorname{ran}(q)$, then χ_A is a branch in $K(\langle p,q\rangle)$. The converse is also true: any branch in $K(\langle p,q\rangle)$ determines the characteristic function of a set separating $\operatorname{ran}(p)$ and $\operatorname{ran}(q)$, so in words $[K(\langle p,q\rangle)] = \mathrm{Sep}_\kappa(\langle p,q\rangle)$. This implies in particular that $\sigma \in \operatorname{ext}(K(\langle p,q\rangle))$ if and only if σ is an initial segment of the characteristic function of a set A separating $\operatorname{ran}(p)$ and $\operatorname{ran}(q)$.

We show that $\operatorname{ext}(K(\langle p,q\rangle))$ is $<\kappa$-closed: let $\lambda < \kappa$ be a limit ordinal and $\sigma \in 2^\lambda$ such that, for all $\alpha \in \lambda$, $\sigma_\alpha = \sigma{\restriction}\alpha \in \operatorname{ext}(K(\langle p,q\rangle))$. Let A be any set separating $\operatorname{ran}(p)$ and $\operatorname{ran}(q)$, and consider the set given by $S = (A \cap \{\alpha \in \kappa \mid \alpha \geq \lambda\}) \cup \{\alpha \in \lambda \mid \sigma(\alpha) = 1\}$. It is clear that σ is an initial segment of χ_S, so we just need to show that S separates $\operatorname{ran}(p)$ and $\operatorname{ran}(q)$. Assume by contradiction that this is not the case, then either $\operatorname{ran}(p) \setminus S \neq \emptyset$ or $\operatorname{ran}(q) \cap S \neq \emptyset$. Assume $\beta \in \operatorname{ran}(p) \setminus S$: if $\beta \geq \lambda$, then

$\beta \notin A$ and therefore $A \notin \mathrm{Sep}_\kappa(\langle p, q\rangle)$. On the other hand, if $\beta < \lambda$, then $\sigma_{\beta+1}(\beta) = 0$ and $\beta \in \mathrm{ran}(p)$, hence $\sigma_\beta \notin \mathrm{ext}(K(\langle p, q\rangle))$. The assumption that $\mathrm{ran}(q) \cap S \neq \emptyset$ leads to a similar contradiction. Hence $\sigma \in \mathrm{ext}(K(\langle p, q\rangle))$ and therefore $\mathrm{ext}(K(\langle p, q\rangle))$ is $<\kappa$-closed.

This implies that the composition $\mathrm{RWKL}_\kappa \circ K$ is well defined, and by our previous discussion we have that $\mathrm{RWKL}_\kappa \circ K(\langle p, q\rangle) = \mathrm{Sep}_\kappa(\langle p, q\rangle)$. Now let K' be a computable realizer of K, we obtian that for any realizer W of RWKL_κ, $W \circ K'$ is a realizer of Sep_κ, therefore $\mathrm{Sep}_\kappa \leq_{\mathrm{sW}} \mathrm{RWKL}_\kappa$. $\square$

We now show that $\widehat{\mathrm{LLPO}_\kappa}$ is enough to compute branches in trees T with $<\kappa$-closed extendible parts, if we have access to a tree oracle (cf. Definition 4.3.4). This proof is a direct translation of the proof of Theorem 8.2 in [6].

Proposition 5.3.4. *Assume Hypothesis 1.2.19 and let o be a tree oracle. We have* $\mathrm{RWKL}_\kappa \leq^o_{\mathrm{W}} \widehat{\mathrm{LLPO}_\kappa}$.

Proof. Given a tree T and an ordinal α, define the sets

$P_{\alpha,0} = \{w \in 2^{<\kappa} \mid \forall v \in T_\alpha \neg(w^\frown 0 \subseteq v \vee v \subseteq w^\frown 0)\}$, $P_{\alpha,1} = \{w \in 2^{<\kappa} \mid \forall v \in T_\alpha \neg(w^\frown 1 \subseteq v \vee v \subseteq w^\frown 1)\}$,

and define the function $m :\subseteq 2^{<\kappa} \to \kappa$ as

$$m(w) = \min\{\alpha \in \kappa \mid w \in P_{\alpha,0} \cup P_{\alpha,1}\}.$$

where $m(w) \uparrow$ if $\{\alpha \in \kappa \mid w \in P_{\alpha,0} \cup P_{\alpha,1}\} = \emptyset$. Notice that m is o-computable. Further, define a function $q\colon 2^{<\kappa} \to 2^\kappa$ as

$$q(w) = \begin{cases} 0^\gamma 1 0^\kappa & \text{if } m(w) \text{ is defined, } w \in P_{m(w),0} \setminus P_{m(w),1}, \text{ and } \gamma \text{ is the } m(w)\text{-th even ordinal,} \\ 0^\gamma 1 0^\kappa & \text{if } m(w) \text{ is defined, } w \in P_{m(w),1} \setminus P_{m(w),0}, \text{ and } \gamma \text{ is the } m(w)\text{-th odd ordinal,} \\ 0^\kappa & \text{otherwise.} \end{cases}$$

Since m is o-computable, we get that q is Type 2 computable with oracle access to o, therefore we can also Type 2 compute the sequence $\langle q(\nu(\alpha))\rangle_{\alpha\in\kappa}$. Exactly as in the proof of [6, Theorem 8.2], we obtain that for all $w \in T$,

$$i \in \mathrm{LLPO}_\kappa(q(w)) \implies \forall \alpha \text{ even } q(w)(\alpha + i) = 0 \implies \exists x \in [T](w^\frown i \subseteq x).$$

In particular this implies that the choice function f given by $w \mapsto w^\frown i$ if and only if $i \in \mathrm{LLPO}_\kappa(q(w))$ has the property that if $w \in T$, then $f(w) \in \mathrm{ext}(T)$. Since $\mathrm{ext}(T)$ is $<\kappa$-closed, we can appeal to Lemma 1.1.38 and obtain that the sequence inductively defined as $r_0 = \emptyset$, $r_{\alpha+1} = f(r_\alpha)$ and $r_\lambda = \bigcup_{\alpha\in\lambda} r_\alpha$ for all limit ordinals λ determines a branch in T. It is clear that such sequence can be computed if we have access to LLPO_κ. This shows that for any $T \in \mathrm{dom}(\mathrm{RWKL}_\kappa)$, we can o-compute the sequence $\langle q(\nu(\alpha))\rangle_{\alpha\in\kappa}$ and use any realizer of $\widehat{\mathrm{LLPO}_\kappa}$ on the latter sequence to obtain an oracle which allows us to compute a branch on T. This shows that $\widehat{\mathrm{LLPO}} \leq^o_{\mathrm{W}} \mathrm{RWKL}_\kappa$. $\square$

We remark that the use of the tree oracle seems to be necessary in a lot of constructions related to trees. Naturally this necessity does not arise in the ω-context. Arguably, tree oracles are used there too, as many constructions (e.g., again Theorem [6, Theorem 8.2], but also any proof of computable compactness of Cantor space, i.e., the classical analogue of our Lemma 4.3.6) rely on deciding statements of the form $P(n) \iff \forall x \in 2^n Q(x)$. These are never mentioned because tree oracles are computable for ω, so we never have to use them as actual oracles.

Now from our discussion in Section 1.1.2, it is very plausible that the function RWKL_κ is stricty below the function WKL_κ. We do not have a proof of this fact, although we have partial results which point in that direction. We present what we currently have.

Proposition 5.3.5. *Assuming Hypothesis 1.2.19, we have* $\mathrm{RWKL}_\kappa \leq_{\mathrm{sW}} \mathrm{Ext}$.

Proof. Given a tree $T \in \mathrm{dom}(\mathrm{RWKL}_\kappa)$, we have that $T \in \mathrm{dom}(\mathrm{Ext})$, hence we can directly apply any realizer H of Ext to a code p for T and obtain a code q for $\mathrm{ext}(T)$. We can then define the computable choice function $f\colon \mathrm{ext}(T) \to \mathrm{ext}(T)$ as

$$f(\sigma) = \begin{cases} \sigma^\frown 0 & \text{if } \sigma^\frown 0 \in \mathrm{ext}(T), \\ \sigma^\frown 1 & \text{otherwise.} \end{cases}$$

Clearly f is computable with oracle access to $\mathrm{ext}(T)$ and again by Lemma 1.1.38 we can define a branch on T by a recursive construction based on f. This shows that, having access to any realizer for Ext, we can compute a realizer for RWKL_κ. Moreover, this realizer only needs access to $\mathrm{ext}(T)$, hence $\mathrm{RWKL}_\kappa \leq_{\mathrm{sW}} \mathrm{Ext}$. □

Proposition 5.3.6. *Let F be the realizer of* Ext. *There is no Type 2 computable function $H :\subseteq \kappa^\kappa \to \kappa^\kappa$ such that the function $p \mapsto H(\langle p, F(p)\rangle)$ realizes* WKL_κ.

Proof. Let $H :\subseteq \kappa^\kappa \to \kappa^\kappa$ be any Type 2 computable function and let p be a code for the tree $T = 2^{<\kappa}$. Assume that $H(\langle p, F(p)\rangle)$ is the characteristic function of a branch $x \in [T]$. Let M be the T2κTM computing H and consider the T2κTM M' which, on input $y \in 2^\kappa$, counts the number of distinct short sequences enumerated by y. In other words, at time step β, the machine M' has read β bits of input and printed 1^δ on the output tape, where $\delta = \mathrm{ot}\{\alpha \in \beta \mid y(\alpha) = 1 \wedge \forall \gamma < \alpha\, (\nu(\alpha) \neq \nu(\gamma))\}$.

By Lemma 1.2.27 let λ be a limit ordinal which is a fixpoint of the time function for the computation of $M' \circ M$ on $\langle p, F(p)\rangle$. By definition of λ and $M' \circ M$, we know that, by step λ, the machine for H has determined the branch x up to at least level λ (because it has found λ-many 1s on the tape for the characteristic function of x, each of them corresponding to a distinct sequence). Therefore, if $q \in \mathrm{dom}(H)$ is such that $\langle p, F(p)\rangle{\restriction}\lambda \subseteq q$, then $x{\restriction}\lambda \subseteq y$ where y is the branch coded by $H(q)$. Now consider the code p' for the tree $T' = 2^{<\kappa} \setminus \{\sigma \in 2^{<\kappa} \mid x{\restriction}\lambda \subseteq \sigma\}$.

It is straightforward to see that both $T' = \mathrm{ext}(T')$ and that these coincide with $T = \mathrm{ext}(T)$ up to level λ. By the bound on the enumeration function expressed in Corollary 1.2.13, we obtain that their characteristic functions also coincide at least up to λ, hence $x{\restriction}\lambda \subseteq H(\langle p', F(p')\rangle)$. Since no branch of T' has $x{\restriction}\lambda$ as an initial segment, it follows that $H(\langle p', F(p')\rangle)$ is not a branch on T'. □

It seems like the proof technique employed in Proposition 5.3.6 cannot be applied to rule out the existence of a pair of functions H, K which witnesses a Weihrauch reduction, therefore a proof of this non-reduction would probably need to follow another route. Nonetheless the result above seems to be a clue that we really do have $\mathrm{WKL}_\kappa \not\leq_{\mathrm{W}} \mathrm{Ext}$. Such result would establish $\mathrm{WKL}_\kappa \not\leq_{\mathrm{W}} \mathrm{RWKL}_\kappa$. Note moreover that the analogous Type 2 computable function $H :\subseteq \omega^\omega \to \omega^\omega$ for the classical context exists and it is essentially the classical analogue of the function described in Proposition 5.3.5 (cf. also Lemma 1.1.36).

We close the section with a few words on the role of $\widehat{\mathrm{LLPO}}_\kappa$ in generalized computable analysis, compared to the role of $\widehat{\mathrm{LLPO}}$ in classical computable analysis. First, the equivalence $\widehat{\mathrm{LLPO}} \equiv_{\mathrm{W}} \mathrm{WKL}$ gives $\widehat{\mathrm{LLPO}}$ a rather important status in the classical context, as Weak Kőnig's Lemma is well-known to be a centerpiece in computable analysis as well as in reverse mathematics (see [20], [7] and [6] for WKL in computable analysis; for reverse mathematics notice that WKL_0 is one of the so-called "big five" subsystems of second order arithmetic [38], which are frameworks that exactly capture the proof theoretic strength of many mathematical statements). Moreover, the classical equivalence $\mathrm{WKL} \equiv_{\mathrm{W}} \mathrm{C_K}$, where $\mathrm{C_K}$ stands for compact choice on the real line, implies that $\widehat{\mathrm{LLPO}}$ is in particular at least as strong as choice on closed and bounded intervals of $\mathbb{R}$.

The matter is not as clear in the generalized context, for multiple reasons: first, we do not know whether $\widehat{\mathrm{LLPO}}_\kappa \equiv_{\mathrm{W}} \mathrm{WKL}_\kappa$ (but we suspect this is not the case). Second, we do not know whether WKL_κ is equivalent to choice on κ-compact subsets of $\mathbb{R}_\kappa$ (we did not investigate this at all due to time constraints). Third, even if the latter equivalence did hold, the set of κ-compact subsets of $\mathbb{R}_\kappa$ does not contain many naturally occurring sets, in particular closed bounded intervals in $\mathbb{R}_\kappa$ are not κ-compact. This suggests that κ-compact choice on $\mathbb{R}_\kappa$ might not be very relevant anyway.

This leaves us with a principle $\widehat{\mathrm{LLPO}}_\kappa$ which is far less important than its classical counterpart in the classification of Weihrauch degrees in generalized computable analysis. This fact is also the reason why the Parallelization Principle is not as useful in the generalized context, as we will see in the next chapter.

5.4 Interval choice principles and boundedness principles

In this section we present the results pertaining to the Weihrauch classification of the interval choice principles $\mathrm{C_{I,c}}$ / $\mathrm{C^-_{I,c}}$ / $\mathrm{C_{I,b}}$ / $\mathrm{C^-_{I,b}}$ and the boundedness principles $\mathrm{B_{I,c}}$ / $\mathrm{B^-_{I,c}}$ / $\mathrm{B_{I,b}}$ / $\mathrm{B^-_{I,b}}$ / $\mathrm{B^+_{I,c}}$ / $\mathrm{B^+_{I,b}}$ / $\mathrm{B_F}$. Most of these were studied by Galeotti either in [16] or in [17].

First we summarize what is already known:

Theorem 5.4.1. *Assume Hypothesis 1.2.19, then we have:*

(a) $\mathrm{IVT}_\kappa \equiv_{\mathrm{W}} \mathrm{B_{I,b}}$,

(b) $\mathrm{C_{I,c}} \equiv_{\mathrm{W}} \mathrm{B_{I,c}}$,

(c) $\mathrm{C^-_{I,c}} \equiv_{\mathrm{W}} \mathrm{B^-_{I,c}}$,

(d) $\mathrm{C_{I,b}} \equiv_{\mathrm{W}} \mathrm{B_{I,b}}$,

(e) $\mathrm{C^-_{I,b}} \equiv_{\mathrm{W}} \mathrm{B^-_{I,b}}$,

(f) $\mathrm{B_F} \equiv_{\mathrm{W}} \mathrm{C}_\kappa$,

(g) $\mathrm{C_{I,c}} \not\leq_{\mathrm{W}} \mathrm{C}_\kappa$,

(h) $\mathrm{C_{I,c}} \leq_{\mathrm{sW}} \mathrm{C_A}$, $\mathrm{C^-_{I,c}} \leq_{\mathrm{sW}} \mathrm{C_A}$ *and* $\mathrm{C_{I,b}} \leq_{\mathrm{sW}} \mathrm{C_A}$.

Proof. Item (a) is [17, Theorem 4.2.4]. Item (b) is stated as a topological Weihrauch equivalence in Proposition 4.5.7 in [16]. We remark that the original proof makes use of the fact that the function $f\colon (0,1) \to \mathbb{R}_\kappa$ defined as:

$$x \mapsto \tfrac{2x-1}{x-x^2}$$

is a strictly increasing homeomorphism between $(0,1)$ and $\mathbb{R}_\kappa$ to transfer the boundedness principle $\mathrm{B_{I,c}}$ from fully general sequences in $\mathbb{R}_\kappa$ to sequences in $(0,1)$. In particular the original proof argues that f^{-1}, being continuous, must have a continuous realizer F (by the Main Theorem of Generalized Computable Analysis) and uses F as one of the two continuous functions witnessing the topological Weihrauch reduction. The effectivization of this step of the proof requires showing that f^{-1} is computable to pick a computable relizer F for it. Luckily, as the referenced proof mentions, we have that for every open $(r,r') \subseteq (0,1)$, $f[(r,r')] = (\frac{2r-1}{r-r^2}, \frac{2r'-1}{r'-r^2})$. Since f is a bijection, it follows that the latter equation is also a way to compute preimages for the function f^{-1}. Applying Theorem 2.2.2 we obtain that f^{-1} is computable, hence we can actually pick the realizer F to be computable. The fact that the rest of the functions involved in the topological Weihrauch reduction are effective is straightforward, therefore we can promote the equivalence to a (computable) Weihrauch equivalence. Items (c), (d) and (e) are witnessed by the same functions witnessing item (b). Item (f) is essentially [16, Proposition 4.5.6]: the result is stated as a topological Weihrauch equivalence in the cited thesis, but the proof is easily seen to be effective, assuming our computable enumeration of $\mathbb{Q}_\kappa$. Item (g) is proven (for topological Weihrauch reducibility) in Proposition 4.5.9 in [16].[4] Item (h) is immediate because all interval choice principles are restrictions of $\mathrm{C_A}$. □

We remark that the proof of items (g) and (h) does not need Hypothesis 1.2.19.

Proposition 5.4.2. *We have* $\mathrm{B_{I,c}} \leq_{\mathrm{sW}} \mathrm{B_{I,b}}$.

[4]Notice that $f \not\leq_{\mathrm{tW}} g$ trivially implies $f \not\leq_{\mathrm{W}} g$.

The proof is immediate as the former is a restriction of the latter, hence the identity function witnesses the reduction. We conjecture that $\mathrm{B}_{\mathrm{I,b}} \leq_{\mathrm{W}} \mathrm{B}_{\mathrm{I,c}}$ does not hold, but we unfortunately could not prove this. We have a weaker result, analogous to Proposition 5.3.6.

Proposition 5.4.3. *There is no continuous function $K :\subseteq \kappa^\kappa \to \kappa^\kappa$ such that for any realizer F of $\mathrm{B}_{\mathrm{I,c}}$, the function $F \circ K$ is a realizer of $\mathrm{B}_{\mathrm{I,b}}$.*

Proof. Let p be a code for $((q_\alpha)_{\alpha\in\kappa}, (q'_\alpha)_{\alpha\in\kappa}) \in \mathbf{S}^{\uparrow}_{\mathrm{b}} \times \mathbf{S}^{\downarrow}_{\mathrm{b}}$ and call $I = \mathrm{B}_{\mathrm{I,b}}((q_\alpha)_{\alpha\in\kappa}, (q'_\alpha)_{\alpha\in\kappa})$ the closed interval of simultaneous bounds that these sequences identify. Suppose that both $(q_\alpha)_{\alpha\in\kappa}$ and $(q'_\alpha)_{\alpha\in\kappa}$ are not convergent. This implies that I does not have either endpoint, and if $K(p)$ belongs to $\mathrm{dom}(F)$, then it must be the case that $K(p)$ codes two different sequences $(s_\alpha)_{\alpha\in\kappa}, (s'_\alpha)_{\alpha\in\kappa} \in \mathbf{S}^{\uparrow}_{\mathrm{c}} \times \mathbf{S}^{\downarrow}_{\mathrm{c}}$ identifying a closed interval of simultaneous bounds I'.

The interval I' will have endpoints (namely the limits of the two sequences) therefore it cannot be the case that $I = I'$. Now if $I' \not\subseteq I$, then let $x \in I' \setminus I$ and pick a realizer F' of $\mathrm{B}_{\mathrm{I,c}}$ which, on input the code $K(p)$ outputs a code q for x. As $x \notin I$, we have that $F' \circ K$ is not a realizer of $\mathrm{B}_{\mathrm{I,b}}$. If on the other hand $I' \subseteq I$ we build another pair of sequences $((t_\alpha)_{\alpha\in\kappa}, (t'_\alpha)_{\alpha\in\kappa})$ as follows. First, notice that $I' \subseteq I$ implies that there exists $\gamma \in \kappa$ such that $s_\gamma \in I$ and $s'_\gamma \in I$. Since K is continuous, there exists $\delta \in \kappa$ such that for any code $q \in \mathrm{dom}(\mathrm{B}_{\mathrm{I,b}} \circ \delta_{\mathbf{S}^{\uparrow}_{\mathrm{b}}} \times \delta_{\mathbf{S}^{\downarrow}_{\mathrm{b}}})$ extending $p{\restriction}\delta$, $K(q)$ codes sequences $((r_\alpha)_{\alpha\in\kappa}, (r'_\alpha)_{\alpha\in\kappa})$ extending $((s_\alpha)_{\alpha\in\gamma+1}, (s'_\alpha)_{\alpha\in\gamma+1})$. In particular this means that the interval of simultaneous bounds identified by $((r_\alpha)_{\alpha\in\kappa}, (r'_\alpha)_{\alpha\in\kappa})$ is contained in $[s_\gamma, s'_\gamma] \subseteq I$. Now let η be an ordinal such that the code $p{\restriction}\delta$ determines an initial segment of length $\leq\eta$ of the sequences $((q_\alpha)_{\alpha\in\kappa}, (q'_\alpha)_{\alpha\in\kappa})$. We extend the code $p{\restriction}\delta$ to a code p' of two strictly monotone sequences $((t_\alpha)_{\alpha\in\kappa}, (t'_\alpha)_{\alpha\in\kappa})$ such that

$$t_\beta = q_\beta,\ t'_\beta = q'_\beta\ \ \forall \beta \leq \eta,$$

$(t_\alpha)_{\alpha\in\kappa}$ converges to $q_{\eta+1}$ and $(t'_\alpha)_{\alpha\in\kappa}$ converges to $q_{\eta+2}$. This implies that $\mathrm{B}_{\mathrm{I,b}}((t_\alpha)_{\alpha\in\kappa}, (t'_\alpha)_{\alpha\in\kappa}) = I'' = [q_{\eta+1}, q_{\eta+2}]$ and $I'' \cap I = \emptyset$. By the argument above, we have that for any realizer F of $\mathrm{B}_{\mathrm{I,c}}$, $F(K(p'))$ codes a κ-real in I'', hence not in I.

This shows that in any case, for any continuous function K, it is impossible that the functions $F \circ K$ realize $\mathrm{B}_{\mathrm{I,b}}$ for all realizers F of $\mathrm{B}_{\mathrm{I,c}}$. □

An analogous proof shows that the same can be said about the relation between $\mathrm{C}_{\mathrm{I,b}}$ and $\mathrm{C}_{\mathrm{I,c}}$.

We summarize here the other straightforward reductions based on domain restrictions:

Proposition 5.4.4. *We have:*

(a) $\mathrm{B}^+_{\mathrm{I,c}} \leq_{\mathrm{sW}} \mathrm{B}^+_{\mathrm{I,b}}$,

(b) $\mathrm{B}^-_{\mathrm{I,c}} \leq_{\mathrm{sW}} \mathrm{B}^-_{\mathrm{I,b}}$,

(c) $\mathrm{B}_{\mathrm{I,b}} \leq_{\mathrm{sW}} \mathrm{B}^+_{\mathrm{I,b}}$,

(d) $\mathrm{B}_{\mathrm{I,c}} \leq_{\mathrm{sW}} \mathrm{B}^+_{\mathrm{I,c}}$,

(e) $\mathrm{B}^-_{\mathrm{I,c}} \leq_{\mathrm{sW}} \mathrm{B}_{\mathrm{I,c}}$,

(f) $\mathrm{B}^-_{\mathrm{I,b}} \leq_{\mathrm{sW}} \mathrm{B}_{\mathrm{I,b}}$.

We omit the proofs as, similarly to Proposition 5.4.2, these reducibilities are witnessed by identity functions. We continue with results analogous to the classical ones:

Proposition 5.4.5. *We have* $\mathrm{B}_{\mathrm{F}} \leq_{\mathrm{sW}} \mathrm{B}^+_{\mathrm{I,c}}$.

Proof. We define a Type 2 computable function H which, for any code p for a strictly increasing sequence of κ-rationals $(q_\alpha)_{\alpha\in\kappa}$ converging to r, returns a code for the pair of sequences $((q_\alpha)_{\alpha\in\kappa}, (+\infty)^\kappa)$. By definition, for any realizer F for $\mathrm{B}^+_{\mathrm{I,c}}$ and any such code p, $F(H(p))$ codes some κ-real greater than or equal to r, therefore the function $F \circ H$ realizes B_{F}. This shows the required strong reduction. □

Proposition 5.4.6. *We have* $\mathrm{B}^-_{\mathrm{I,c}} \leq_{\mathrm{W}} \mathrm{B}_{\mathrm{F}}$.

Proof. We follow the classical proof (see [7, Proposition 3.8]). Suppose we are given a name $p \in \kappa^\kappa$ coding two sequences $(q_\alpha)_{\alpha\in\kappa} \in \mathbf{S}^{\uparrow}_{\mathrm{c}}$ and $(q'_\alpha)_{\alpha\in\kappa} \in \mathbf{S}^{\uparrow}_{\mathrm{c}}$ such that

$$\lim_{\alpha\in\kappa} = q < q' = \lim_{\alpha\in\kappa} q'_\alpha.$$

We build a machine M that works as follows: the machine reserves a tape T_i to use as an ordinal register, initialized to 0. The machine will in the long run produce a sequence of guesses given by midpoints of the form $(q_\alpha + q'_\alpha)/2$, as well as a sequence of ordinals which is meant to keep track of the changes to the sequence of midpoints. At first, M computes the midpoint

$$c_0 = \tfrac{q_0+q'_0}{2}.$$

It then starts parsing the sequences $(q_\alpha)_{\alpha\in\kappa}$ and $(q'_\alpha)_{\alpha\in\kappa}$ and for all α it checks whether the current guess c satisfies $q_\alpha < c < q'_\alpha$. If this is the case, the machine prints a $\delta_{\mathbb{Q}_\kappa}$-code for the ordinal currently contained in T_i and it goes on to performing the same check for the ordinal $\alpha + 1$. If this is not the case, then the machine prints a $\delta_{\mathbb{Q}_\kappa}$-code for α, and it computes the new guess $(q_\alpha + q'_\alpha)/2$, before moving to the next ordinal.

We remark that the content of the register T_i behaves correctly at limit times as it may only increase with applications of the lim sup rule. We show that $M(p)$ codes a bounded sequence of ordinals. Let $\epsilon = q' - q$ and, since both sequences $(q_\alpha)_{\alpha\in\kappa}$ and $(q'_\alpha)_{\alpha\in\kappa}$ are convergent, let γ be such that, for all $\eta \geq \gamma$, $q - q_\eta \leq \epsilon/4$ and $q'_\eta - q' \leq \epsilon/4$. We obtain that for all $\eta \geq \gamma$:

$$\tfrac{1}{2}\epsilon \leq \tfrac{q'_\eta - q_\eta}{2} \leq \tfrac{3}{4}\epsilon,$$

therefore

$$q \leq c_{\eta,\eta} = q_\eta + \tfrac{q'_\eta - q_\eta}{2} \leq q'.$$

This shows that no ordinal greater than or equal to $\eta + 1$ is ever reached by the sequence coded by $M(p)$. Call $K :\subseteq \kappa^\kappa \to \kappa^\kappa$ the Type 2 computable function computed by M. The argument above shows that, for every $p \in \mathrm{dom}(\mathrm{B}^-_{\mathrm{I,c}} \circ \delta_{\mathbf{S}^{\uparrow}_{\mathrm{c}}} \times \delta_{\mathbf{S}^{\downarrow}_{\mathrm{c}}})$, $K(p)$ codes an eventually constant, hence convergent sequence of ordinals (seen as κ-rationals) such that, if $\beta \in \kappa$ is an upper bound for it, then we can be sure that the center which is guessed at stage β in the computation of M on p actually lies between q and q'. We then use a realizer F of C_κ to determine a κ-real x upper bound for $K(p)$ and we compute some ordinal $\beta > x$ (notice that this is trivially computable from a $\delta_{\mathbb{R}_\kappa}$-name of x). Finally we simulate the computation of M on p up to stage β and we output a $\delta_{\mathbb{R}_\kappa}$-name for the center guessed by M at stage β. Call the function associated to this procedure H. This shows that, given a realizer F for B_{F}, the function $p \mapsto H(\langle p, F(K(p))\rangle)$ is a realizer of $\mathrm{B}^-_{\mathrm{I,c}}$, as desired. □

We now state a new result, which, in the classical context, is a consequence of some reductions which are not currently proven in the generalized context. Nonetheless, it admits a straightforward, direct proof, and it is at the moment the best we can obtain.

Proposition 5.4.7. *We have* $\mathrm{B}^+_{\mathrm{I,c}} <_{\mathrm{W}} \mathrm{B}$.

Proof. Notice that for every pair of sequences $((q_\alpha)_{\alpha\in\kappa}, (q'_\alpha)_{\alpha\in\kappa}) \in \mathrm{dom}(\mathrm{B}^+_{\mathrm{I,c}})$, the left sequence $(q_\alpha)_{\alpha\in\kappa}$ is guaranteed to be a stricty increasing sequence converging to the real $r \in \mathbb{R}_\kappa$. Therefore, given a code p for the pair of sequences $((q_\alpha)_{\alpha\in\kappa}, (q'_\alpha)_{\alpha\in\kappa})$ we can computably extract a code p' for $(q_\alpha)_{\alpha\in\kappa}$ and, given any realizer F for B, we can compute a $\delta_{\mathbb{R}_\kappa}$-name for its limit r. By definition, $r \in \mathrm{B}^+_{\mathrm{I,c}}((q_\alpha)_{\alpha\in\kappa}, (q'_\alpha)_{\alpha\in\kappa})$, so this means that we can compute a realizer for $\mathrm{B}^+_{\mathrm{I,c}}$ from any realizer of B. This shows the required reduction.

To show that $\mathrm{B} \not\leq_{\mathrm{W}} \mathrm{B}^+_{\mathrm{I,c}}$, we prove that $\mathrm{EC}_\kappa \not\leq_{\mathrm{W}} \mathrm{B}^+_{\mathrm{I,c}}$. Since there are κ-computably enumerable sets $A \subseteq \kappa$ which are not computable, it follows that EC_κ does not map computable inputs to computable outputs. On the other hand, we claim that $\mathrm{B}^+_{\mathrm{I,c}}$ admits a realizer which maps computable inputs to computable outputs. Notice that if $((q_\alpha)_{\alpha\in\kappa}, (q'_\alpha)_{\alpha\in\kappa}) \in \mathrm{dom}(\mathrm{B}^+_{\mathrm{I,c}})$, $q = \lim_{\alpha\in\kappa} q_\alpha$ and

$q' = \lim_{\alpha\in\kappa} q'_\alpha$, there are two possibilities: if $q \neq q'$, then the interval (q, q') contains a κ-rational, hence a computable point. If on the other hand $q = q'$ and (the codes for) the sequences $((q_\alpha)_{\alpha\in\kappa}$ and $(q'_\alpha)_{\alpha\in\kappa})$ are computable, it immediately follows that q has a computable $\delta_{\mathbb{R}_\kappa}$-name. This implies that there exists a realizer of $\mathrm{B}^+_{\mathrm{I,c}}$ which maps computable inputs to computable outputs. By Corollary 4.1.3 it follows that $\mathrm{EC}_\kappa \not\leq_{\mathrm{W}} \mathrm{B}^+_{\mathrm{I,c}}$. $\square$

As in the classical setting (cf. [7, Theorem 5.2]), the Weihrauch degree of BCT_κ is captured by C_κ and B_F.

Proposition 5.4.8. *For any κ-spherically complete computable κ-metric space $\mathfrak{X} = (X, d, s)$, we have* $\mathrm{BCT}_\kappa(\mathfrak{X}) \equiv_{\mathrm{W}} \mathrm{C}_\kappa \equiv_{\mathrm{W}} \mathrm{B}_\mathrm{F}$.

Proof. We follow the proof referenced. We start by showing $\mathrm{BCT}_\kappa(\mathfrak{X}) \leq_{\mathrm{W}} \mathrm{C}_\kappa$. Let p be a $(\boldsymbol{\Pi}^0_1(\mathfrak{X}))^\kappa$-name for a sequence of closed sets $(A_\alpha)_{\alpha\in\kappa}$ such that $X = \bigcup_{\alpha\in\kappa} A_\alpha$. Recall that this means that each p_α is an enumeration for codes of basic open balls of X exhausting the set $X \setminus A_\alpha$. Now for all $\alpha \in \kappa$, define $B_\alpha = B_X(s(\beta), \overline{\gamma})$ where $\alpha = \ulcorner \beta, \gamma \urcorner$. The sequence $(B_\alpha)_{\alpha\in\kappa}$ is an enumeration of all basic open balls of X. Consider the set P given by

$$P = \{\ulcorner \alpha, \beta \urcorner \mid (X \setminus A_\alpha) \cap B_\beta \neq \emptyset \vee B_\beta = \emptyset\}.$$

It is clear that, having access to p, we can compute a sequence $q \in \kappa^\kappa$ such that $\mathrm{ran}(q) = P$,[5] in other words, q is a $\delta'_{\boldsymbol{\Pi}^0_1(\kappa)}$-name for $\kappa \setminus P$. Notice that if $\ulcorner \beta, \gamma \urcorner \in \kappa \setminus P$, it follows that $B_\gamma \neq \emptyset$ and $B_\gamma \subseteq A_\beta$. This implies that we can use a realizer of C_κ to compute an index β for a closed set A_β in the original list with nonempty interior. This proves that $\mathrm{BCT}_\kappa(\mathfrak{X}) \leq_{\mathrm{W}} \mathrm{C}_\kappa$. We now prove that $\mathrm{B}_\mathrm{F} \leq_{\mathrm{W}} \mathrm{BCT}_\kappa(\mathfrak{X})$, again by following the proof in [7]. Given a code p for a sequence $(q_\alpha)_{\alpha\in\kappa} \in \mathbf{S}^\uparrow_\mathrm{c}$, define the sequence of closed sets $(A_\alpha)_{\alpha\in\kappa}$ as

$$A_\alpha = \begin{cases} \emptyset & \text{if } \exists\beta(\alpha < q_\beta), \\ X & \text{otherwise.} \end{cases}$$

We explain how this is done computably: for every α, we parse the sequence $(q_\alpha)_{\alpha\in\kappa}$ and we compare it to the κ-rational α. While doing so, we print codes for the empty open ball. If ever we find an ordinal β such that $\alpha < q_\beta$, we start enumerating every basic open ball of X. This leads us to print a $\boldsymbol{\Sigma}^0_1$-name for $\emptyset$ if the condition $\alpha < q_\beta$ is never satisfied, and a $\boldsymbol{\Sigma}^0_1$-name for X otherwise. Hence, we print the correct $\boldsymbol{\Pi}^0_1$-names in both cases. Notice that the closed set A_α has nonempty interior if and only if the κ-rational α is an upper bound to the sequence $(q_\alpha)_{\alpha\in\kappa}$. This implies that, using a realizer of $\mathrm{BCT}_\kappa(\mathfrak{X})$, we can compute a realizer for B_F and thus it shows that $\mathrm{B}_\mathrm{F} \leq_{\mathrm{W}} \mathrm{BCT}_\kappa(\mathfrak{X})$. $\square$

Corollary 5.4.9. *We have* $\mathrm{B}_\mathrm{F} <_{\mathrm{W}} \mathrm{B}^+_{\mathrm{I,c}}$.

Proof. By Proposition 5.4.5 we just need to prove that $\mathrm{B}^+_{\mathrm{I,c}} \not\leq_{\mathrm{W}} \mathrm{B}_\mathrm{F}$. We do this by contradiction, if this were the case, then we would have $\mathrm{B}_\mathrm{F} \equiv_{\mathrm{W}} \mathrm{B}^+_{\mathrm{I,c}}$ and by item (*d*) in Proposition 5.4.4 we would obtain $\mathrm{B}_{\mathrm{I,c}} \leq_{\mathrm{W}} \mathrm{B}_\mathrm{F}$, or equivalently, by Proposition 5.4.8 and item (*b*) in Theorem 5.4.1, $\mathrm{C}_{\mathrm{I,c}} \leq_{\mathrm{W}} \mathrm{C}_\kappa$ which contradicts item (*g*) in Theorem 5.4.1. $\square$

We close the section by stating two more reductions, which are again proved in a way that is analogous to their classical counterpart (cf. [7, Proposition 3.]). These concern the relation between the omniscience principles LLPO_κ and LPO_κ and the generalized choice and boundedness principles.

Proposition 5.4.10. *We have* $\mathrm{LLPO}_\kappa <_{\mathrm{W}} \mathrm{B}^-_{\mathrm{I,c}}$ *and* $\mathrm{LPO}_\kappa <_{\mathrm{W}} \mathrm{B}_\mathrm{F}$.

Proof. We first show $\mathrm{LPO}_\kappa \leq_{\mathrm{W}} \mathrm{B}_\mathrm{F}$: given a sequence $p \in \kappa^\kappa$, we compute (a code for) the sequence of κ-rationals $(q_\alpha)_{\alpha\in\kappa}$ given by $q_\alpha = 0$ if $p(\beta) \neq 0$ for all ordinals $\beta < \alpha$ and, if we find that $p(\delta) = 0$ and δ is the least such ordinal, then $q_\alpha = \delta$ for all $\alpha \geq \delta$. It is clear that the sequence $(q_\alpha)_{\alpha\in\kappa}$ is eventually constant, hence it is in $\mathrm{dom}(\mathrm{B}_\mathrm{F})$. Moreover, if $q \in \mathrm{B}_\mathrm{F}((q_\alpha)_{\alpha\in\kappa})$, we can compute the least ordinal β

[5]Given $\gamma = \ulcorner \alpha, \beta \urcorner$, we print γ on the tape as soon as (if ever) we find out that $X \setminus A_\alpha \neq \emptyset$ or $B_\beta = \emptyset$. Notice that each of these checks is semidecidable.

such that $\beta \geq q$ and use β to compute $\mathrm{LPO}(p)$: by the construction of $(q_\alpha)_{\alpha\in\kappa}$ we know that, if there is a 0 on the sequence p, then it must appear before position β. This shows $\mathrm{LPO}_\kappa \leq_{\mathrm{W}} \mathrm{B_F}$.

The fact that $\mathrm{B_F} \not\leq_{\mathrm{W}} \mathrm{LPO}_\kappa$ follows from the Mind Change Principle: no limit κ-Turing machine can compute $\mathrm{B_F}$ with just one mind change, while computing LPO_κ with one mind change is possible (notice that the latter statement was essentially proven in Proposition 4.2.2).

We now prove $\mathrm{LLPO}_\kappa \leq_{\mathrm{W}} \mathrm{B}^-_{\mathrm{I,c}}$, again following the referenced proof. Given $p \in \mathrm{LLPO}_\kappa$ we start producing codes for sequences $(q_\alpha)_{\alpha\in\kappa} \in \mathbf{S}^{\uparrow}_{\mathrm{c}}$ and $(q_\alpha)_{\alpha\in\kappa} \in \mathbf{S}^{\downarrow}_{\mathrm{c}}$ given by $q_\alpha = -\frac{1}{\alpha+1}$ and $q'_\alpha = 1 + \frac{1}{\alpha+1}$. If we ever find a 1 on p on cell δ and δ is even, we keep on computing codes for $q_\alpha = -\frac{1}{\alpha+1}$ but we switch the right sequence to $q'_\alpha = \frac{1}{4} + \frac{1}{\alpha+1}$ for all $\alpha \geq \delta$. Similarly, if δ is odd, we keep on computing codes for $q'_\alpha = 1+\frac{1}{\alpha+1}$ but we switch the left sequence to $q_\alpha = \frac{3}{4} - \frac{1}{\alpha+1}$. Call H the Type 2 computable function defined by the procedure described above. We have that for any realizer F of $\mathrm{B_{I,c}}$, if $F(H(p))$ codes a κ-real in $[0, \frac{1}{4}]$ then $1 \in \mathrm{LLPO}_\kappa(p)$ and if it codes a κ-real in $[\frac{3}{4}, 1]$, then $0 \in \mathrm{LLPO}_\kappa$. If it codes some other κ-real (notice that by construction this real must always be in $[0, 1]$), then both 0 and 1 are in $\mathrm{LLPO}_\kappa(p)$. This shows that we can compute a realizer of LLPO_κ with access to one for $\mathrm{B_{I,c}}$, in other words, that $\mathrm{LLPO}_\kappa \leq_{\mathrm{W}} \mathrm{B_{I,c}}$. As above, the existence of a reduction in the other direction is ruled out by the Mind Change Principle. □

We conclude the section by proving that, again in analogy with the classical case, closed choice is at least as strong as the strongest interval boundedness principle $\mathrm{B}^+_{\mathrm{I,b}}$.

Proposition 5.4.11. *We have* $\mathrm{B}^+_{\mathrm{I,b}} \leq_{\mathrm{W}} \mathrm{C_A}$.

Proof. Let M be the T2κTM which, on input a code p for the pairs of sequences $(q_\alpha)_{\alpha\in\kappa}$ and $(q'_\alpha)_{\alpha\in\kappa}$, enumerates codes for all open balls in

$$\bigcup_{\alpha\in\kappa}(-\infty, q_\alpha) \cup \bigcup_{\alpha\in\kappa}(q'_\alpha, +\infty).$$

Let H be the Type 2 computable function defined by M. It is straightforward to see that, if F is any realizer of $\mathrm{C_A}$, then $F \circ H$ realizes $\mathrm{B}^+_{\mathrm{I,b}}$. This proves the required reduction. □

Chapter 6

Conclusions

We present a summary of the classification results achieved, together with a comparison between the portion of the Weihrauch hierarchy studied in [7] and the corresponding portion of the Weihrauch hierarchy obtained in the thesis. We also collect open questions and potential directions for future work.

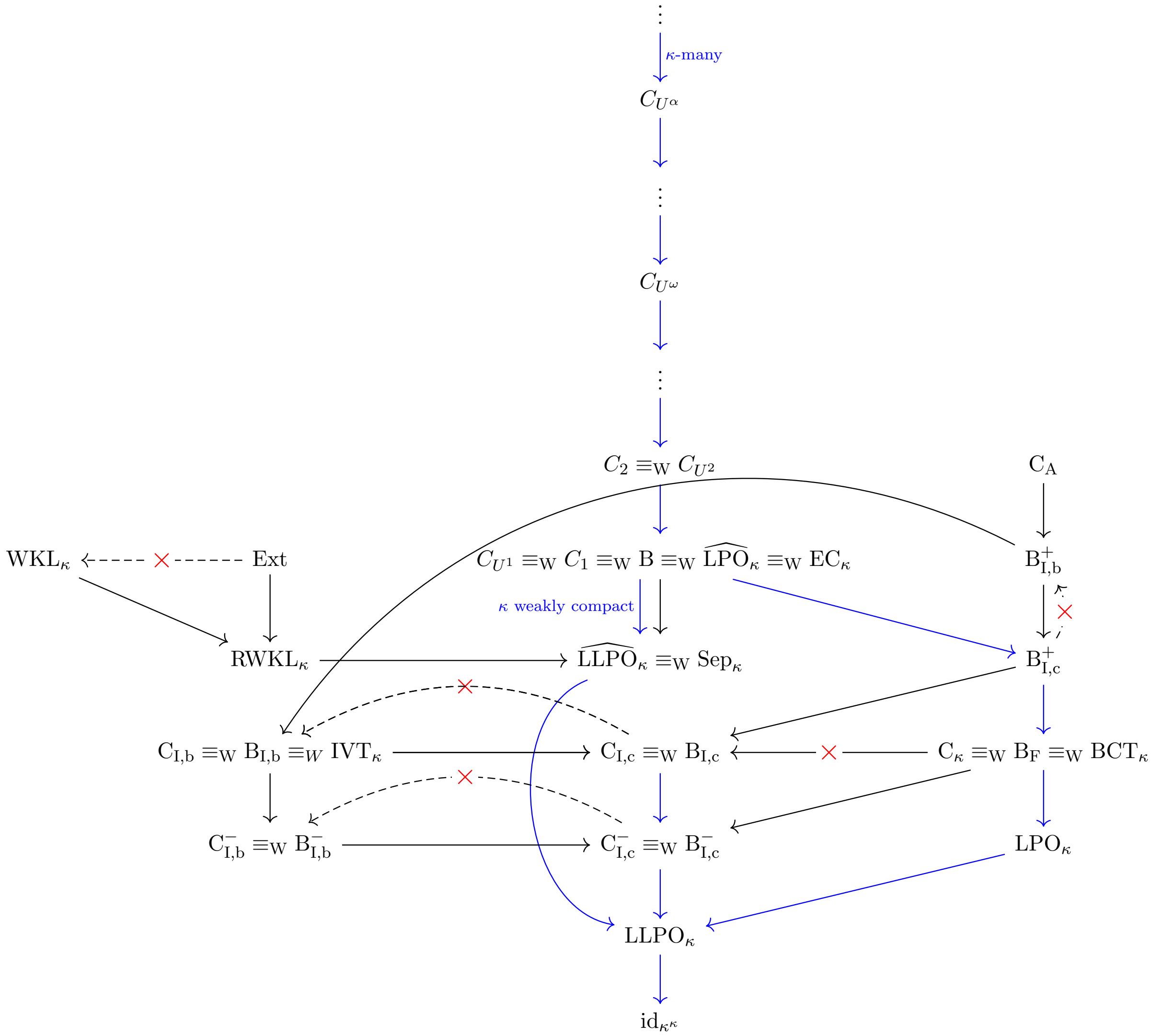
6.1 Comparison with the classical picture

We again propose the table summarising the classification results in [7].

$$
\begin{tikzcd}
\vdots \arrow[d] & \\
C_2 \arrow[d] & \\
C_1 \equiv_{\mathrm{W}} \mathrm{B}_{\mathbb{R}} \equiv_{\mathrm{W}} \widehat{\mathrm{LPO}} \equiv_{\mathrm{W}} \mathrm{EC} \arrow[d] & \\
\mathrm{C}_{\mathrm{A}} \arrow[d] \arrow[ddr, bend left] & \\
\mathrm{C}_{\mathrm{K}} \equiv_{\mathrm{W}} \widehat{\mathrm{LLPO}} \equiv_{\mathrm{W}} \mathrm{WKL} \equiv_{\mathrm{W}} \mathrm{Sep} \arrow[d] & \\
\mathrm{C}_{\mathrm{I}} \equiv_{\mathrm{W}} \mathrm{B}_{\mathrm{I}} \equiv_{\mathrm{W}} \mathrm{IVT} \arrow[d] & \mathrm{B}_{\mathrm{I}}^{+} \arrow[l] \arrow[d] \\
\mathrm{C}_{\mathrm{I}}^{-} \equiv_{\mathrm{W}} \mathrm{B}_{\mathrm{I}}^{-} \arrow[d] & \mathrm{C}_{\omega} \equiv_{\mathrm{W}} \mathrm{B}_{\mathrm{F}} \equiv_{\mathrm{W}} \mathrm{BCT} \arrow[l] \arrow[d] \\
\mathrm{LLPO} \arrow[d] & \mathrm{LPO} \arrow[l] \\
\mathrm{id}_{\omega^\omega} &
\end{tikzcd}
$$

All symbols stand for principles on the real line and on Baire/Cantor space analogous to those of the same name defined for $\mathbb{R}_\kappa$, κ^κ and 2^κ in Chapter 3. Here, an arrow from principle P to P' indicates that $P' <_{\mathrm{W}} P$. As we mentioned in the introduction, all these reductions are either proved or referenced in [7], with the exception of $\mathrm{EC} \equiv_{\mathrm{W}} \mathrm{B}_{\mathbb{R}}$, which is proved in [39, Theorem 4] and $\mathrm{Sep} \equiv_{\mathrm{W}} \mathrm{WKL}$, which is proved in [20, Theorem 6.7]. Moreover, Brattka and Gherardi prove in [7] that the diagram is complete, i.e., no arrow can be added except those which follow from transitivity. Lastly, by [5, Theorem 7.6], each of the C_ns is Weihrauch complete for the class of $\mathbf{\Sigma}^0_{n+1}$-effectively measurable single valued total functions between computable metric spaces.

On the other hand, the following is the table containing all classification results in this thesis. We focus on results pertaining to ordinary Weihrauch reducibility $\leq_{\mathrm{W}}$, and we assume Hypothesis 1.2.19 as many results are obtained under the assumption of computable enumerability of 2^κ. Since the table is meant to capture all the currently known results, we do not distinguish results which need Hypothesis 1.2.19 from those who hold unconditionally in the picture. For more details we refer to the proofs of these results in the thesis.



Where a black arrow from P to P' indicates that $P' \leq_{\mathrm{W}} P$, a blue arrow indicates that $P' <_{\mathrm{W}} P$, a crossed out black arrow indicates that $P' \nleq_{\mathrm{W}} P$, and lastly a dashed and crossed out arrow indicates that we conjecture $P' \nleq_{\mathrm{W}} P$ but we don't have a proof. Again we omitted arrows and non-arrows which follow from transitivity considerations. Notice that, besides arrow explicitly forbidden (the reverse directions of blue arrows and crossed out black arrows) and those forbidden by transitivity considerations, we do not know whether it is possible to add arrows to the table above.

We give references to the results summarized in the picture.
The equivalence $\mathrm{B} \equiv_W \mathrm{EC}_\kappa \equiv_{\mathrm{W}} \widehat{\mathrm{LPO}}_\kappa$ is Corollary 5.2.16, the equivalence $\mathrm{BCT}_\kappa(\mathfrak{X}) \equiv_{\mathrm{W}} \mathrm{C}_\kappa \equiv_{\mathrm{W}} \mathrm{B_F}$ is Proposition 5.4.8 (notice that the equivalence $\mathrm{C}_\kappa \equiv_{\mathrm{W}} \mathrm{B_F}$ is proved in [16] as mentioned in item (f) of Theorem 5.4.1). The equivalence $\widehat{\mathrm{LLPO}}_\kappa \equiv_{\mathrm{W}} \mathrm{Sep}_\kappa$ is Proposition 5.3.1. The equivalences $\mathrm{C_{I,c}} \equiv_{\mathrm{W}} \mathrm{B_{I,c}}$, $\mathrm{C}^-_{\mathrm{I,c}} \equiv_{\mathrm{W}} \mathrm{B}^-_{\mathrm{I,c}}$, $\mathrm{C_{I,b}} \equiv_{\mathrm{W}} \mathrm{B_{I,b}} \equiv_{\mathrm{W}} \mathrm{IVT}_\kappa$ and $\mathrm{C}^-_{\mathrm{I,b}} \equiv_{\mathrm{W}} \mathrm{B}^-_{\mathrm{I,b}}$ are items (a)-(e) in Theorem 5.4.1.

The strictly descending chain of $(C_n)_{n\in\omega}$ is established in Theorem 2.3.12 and Corollary 2.3.14. Moreover, Theorem 2.3.12 yields that each C_n is Weihrauch complete for the set of $\mathbf{\Sigma}^0_{n+1}$-computable partial functions on κ^κ. The chain $(C_{U^\alpha})_{\alpha\in\kappa}$ is given by Galeotti's functions, which we mentioned below Corollary 2.3.14. The Weihrauch reduction $\widehat{\mathrm{LLPO}}_\kappa \leq_{\mathrm{W}} \widehat{\mathrm{LPO}}_\kappa$ follows from Proposition 4.3.2

together with the monotonicity of parallelization (Proposition 1.3.18, item (*b*)). In case κ is weakly compact, this reduction is strict, as stated in Corollary 4.3.21. The strict reduction $\mathrm{B}^+_{\mathrm{I,c}} <_{\mathrm{W}} \mathrm{B}$ is proven in Proposition 5.4.7. The reductions $\mathrm{B}^+_{\mathrm{I,c}} \leq_{\mathrm{W}} \mathrm{B}^+_{\mathrm{I,b}}$, $\mathrm{B}_{\mathrm{I,c}} \leq_{\mathrm{W}} \mathrm{B}_{\mathrm{I,b}}$, $\mathrm{B}^-_{\mathrm{I,c}} \leq_{\mathrm{W}} \mathrm{B}^-_{\mathrm{I,b}}$, $\mathrm{B}_{\mathrm{I,b}} \leq_{\mathrm{W}} \mathrm{B}^+_{\mathrm{I,b}}$, $\mathrm{B}_{\mathrm{I,c}} \leq_{\mathrm{W}} \mathrm{B}^+_{\mathrm{I,c}}$, $\mathrm{B}^-_{\mathrm{I,c}} \leq_{\mathrm{W}} \mathrm{B}_{\mathrm{I,c}}$ and $\mathrm{B}^-_{\mathrm{I,b}} \leq_{\mathrm{W}} \mathrm{B}_{\mathrm{I,b}}$, which all boil down to domain restrictions, are stated in Proposition 5.4.2 and Proposition 5.4.4. The non-reduction $\mathrm{C}_{\mathrm{I,c}} \not\leq_{\mathrm{W}} \mathrm{C}_\kappa$ is item (*g*) in Theorem 5.4.1. The strict reduction $\mathrm{B}_{\mathrm{F}} <_{\mathrm{W}} \mathrm{B}^+_{\mathrm{I,c}}$ is Corollary 5.4.9. The reduction $\mathrm{B}^-_{\mathrm{I,c}} \leq_{\mathrm{W}} \mathrm{C}_\kappa$ is proved in Proposition 5.4.6. The strict reductions $\mathrm{LPO}_\kappa <_{\mathrm{W}} \mathrm{B}_{\mathrm{F}}$ and $\mathrm{LLPO}_\kappa <_{\mathrm{W}} \mathrm{B}^-_{\mathrm{I,c}}$ are Proposition 5.4.10. The reduction $\mathrm{LLPO}_\kappa \leq_{\mathrm{W}} \mathrm{LPO}_\kappa$ is Proposition 4.3.2 and, as we remarked below the proof of the proposition mentioned, its strictness follows from an adaptation of the proof of [40, Theorem 4.2]. The reduction $\mathrm{id}_{\kappa^\kappa} <_{\mathrm{W}} \mathrm{LLPO}_\kappa$ follows from the fact that $\mathrm{id}_{\kappa^\kappa}$ reduces to any other function with a computable point in its domain, and its strictness is item (*e*) in Proposition 5.1.1. The reduction $\mathrm{RWKL}_\kappa \leq_{\mathrm{W}} \mathrm{WKL}_\kappa$ follows by domain restriction, whereas the reductions $\mathrm{RWKL}_\kappa \leq_{\mathrm{W}} \mathrm{Ext}$ and $\mathrm{Sep}_\kappa \equiv_{\mathrm{W}} \mathrm{RWKL}_\kappa$ are Proposition 5.3.5 and Lemma 5.3.3 respectively. The conjectured $\mathrm{WKL}_\kappa \not\leq_{\mathrm{W}} \mathrm{Ext}$ is discussed in Proposition 5.3.6 and lastly the conjectured non-reductions $\mathrm{B}_{\mathrm{I,b}} \not\leq \mathrm{B}_{\mathrm{I,c}}$, $\mathrm{B}^-_{\mathrm{I,b}} \not\leq \mathrm{B}^-_{\mathrm{I,c}}$ and $\mathrm{B}^+_{\mathrm{I,b}} \not\leq \mathrm{B}^+_{\mathrm{I,c}}$ are discussed in Proposition 5.4.3.

Comparing the two tables, we immediately see that all the reductions which hold in the generalized context also hold in the classical context. The main differences between the two are: first, we do not know whether the interval choice principles are reducible to $\widehat{\mathrm{LLPO}_\kappa}$ (but we have reasons to think this is not the case, cf. the discussion at the end of Section 5.3). This makes us unable to exploit the Parallelization Principle, which is the tool that Brattka employed to rule out arrows going from principles below $\widehat{\mathrm{LLPO}}$ to any other principle not below it in the classical table. As we explained at the end of Section 5.3, the way the reduction $\mathrm{C}_{\mathrm{I}} \leq_{\mathrm{W}} \widehat{\mathrm{LLPO}}$ is proven in the classical setting is through compact choice C_{K} and WKL. As mentioned there, this route is unavailable in the generalized context. Nonetheless, it would still be interesting to settle the status of WKL_κ in relation to $\widehat{\mathrm{LLPO}_\kappa}$. Notice that a proof of $\mathrm{WKL}_\kappa \not\leq_{\mathrm{W}} \mathrm{Ext}$ would suffice to obtain that $\widehat{\mathrm{LLPO}_\kappa}$ and WKL_κ have distinct Weihrauch degrees. Second, we observe that each boundedness principle (and the corresponding interval choice principle) is split in two a priori different principles, following the fact that not all strictly monotone bounded sequences in $\mathbb{R}_\kappa$ are convergent. The equivalence $\mathrm{IVT}_\kappa \equiv_{\mathrm{W}} \mathrm{B}_{\mathrm{I,b}} \equiv_{\mathrm{W}} \mathrm{C}_{\mathrm{I,b}}$ is an inkling to the fact that the bounded versions of these principles might turn out to be more useful than their convergent counterparts in the classification of generalized analysis theorems. For this reason we think that it would be interesting to settle the matter of whether the Weihrauch degrees of the bounded principles are actually distinct from the degrees of the corresponding convergent principles.

6.2 Future work

We collect here the questions which have come up during the writing of this thesis and are currently still open, as well as some ideas on how to further develop the results in this thesis.

Open Question 6.2.1. *Does the analogue of the Representation Theorem [5, Theorem* 6.1*] hold for computable κ-metric spaces? In other words, is it the case that a total function between computable κ-metric spaces is $\mathbf{\Sigma}^0_n$-effectively measurable if and only if it has a $\mathbf{\Sigma}^0_n$-effectively measurable realizer?*

As mentioned in Section 2.3 a positive answer to this question would immediately yield a strengthening of our Completeness Theorem (Theorem 2.3.12). Moreover, the Representation Theorem for computable κ-metric spaces would be evidence that, also in the generalized context, the notion of $\mathbf{\Sigma}^0_n$-effective measurability is a natural one for computable analysis.

Open Question 6.2.2. *Does* $\widehat{\mathrm{LPO}_\kappa} \not\leq_{\mathrm{tW}} \widehat{\mathrm{LLPO}_\kappa}$ *hold without the assumption of weak compactness of the cardinal κ?*

Open Question 6.2.3. *Is it the case that* $\mathrm{WKL}_\kappa \not\leq \mathrm{Ext}$*?*

We mention that the proof of Kreisel's Basis Lemma (see [31, Proposition V.5.31]), which states that any computable tree has a Δ^0_2-branch, essentially relies on the consideration that for any computable tree T, the subtree $\mathrm{ext}(T)$ is Δ^0_2 and there exists a computable function $H :\subseteq \omega^\omega \to \omega^\omega$ such that $H\langle \chi_T, \chi_{\text{ext}(T)}\rangle = \chi_b$ where $b \in [T]$ for all trees $T \subseteq 2^{<\omega}$ (cf. Example 1.1.36). Since we know that by Proposition 5.3.6, such function H does not exist in the κ-context, this raises the further question of determining a Basis result for binary trees (with branches) of height κ.

Open Question 6.2.4. *Is it the case that the boundedness principles actually split, i.e., do the non-reductions* $\mathrm{B_{I,b}} \not\leq \mathrm{B_{I,c}}$, $\mathrm{B^-_{I,b}} \not\leq \mathrm{B^-_{I,c}}$ *and* $\mathrm{B^+_{I,b}} \not\leq \mathrm{B^+_{I,c}}$ *hold?*

Lastly we remark that some non-reductions in the classical table, such as $C_1 \not\leq \mathrm{C_A}$, $\mathrm{C_A} \not\leq \mathrm{B^+_{I,b}}$ and $\mathrm{C_K} \not\leq \mathrm{C_I}$ (see [7, Proposition 4.6] and [7, Proposition 4.8]) rely on *recursion theoretic knowledge* about the real line $\mathbb{R}$. In particular, the former relies on the fact that any non-empty closed subset of $\mathbb{R}$ with a computable $\mathbf{\Pi}^0_1$-name has at least a point x with a *low name* i.e., such that there exists $p \in \omega^\omega$ with $p \leq_{\mathrm{T}} \emptyset'$ and $\delta_{\mathbb{R}}(p) = x$, whereas the latter two rely on the fact that there exists a nonempty compact set $A \subseteq \mathbb{R}$ which admits a computable $\mathbf{\Pi}^0_1$-name and has no computable points. At the moment, no result of this type is known for $\mathbb{R}_\kappa$, but we think that this could be a good direction of investigation, both for independent interest, and to be able to exploit the Turing degree invariance principle to obtain more non-reductions between our generalized principles.

Bibliography

[1] C. Agostini, L. Motto Ros, and P. Schlicht. Generalized polish spaces at regular uncountable cardinals, 2021. Preprint arXiv:2107.02587.

[2] N.L. Alling. *Foundations of Analysis over Surreal Number Fields*, volume 141 of *North-Holland Mathematics Studies*. Elsevier Science, 1987.

[3] D. Asperó and K. Tsaprounis. Long reals. *Journal of Logic and Analysis*, 10(1):1–36, 2018.

[4] V. Brattka. *Computability of Banach space principles.* Number 286 in Informatik-Berichte. FernUniversität in Hagen, 2001.

[5] V. Brattka. Effective Borel measurability and reducibility of functions. *Mathematical Logic Quarterly*, 51(1):19–44, 2005.

[6] V. Brattka and G. Gherardi. Weihrauch degrees, omniscience principles and weak computability. *Journal of Symbolic Logic*, 76(1):143–176, 2009.

[7] V. Brattka and G. Gherardi. Effective choice and boundedness principles in computable analysis. *Bulletin of Symbolic Logic*, 17(1):73–117, 2011.

[8] V. Brattka and G. Presser. Computability on subsets of metric spaces. *Theoretical Computer Science*, 305(1):43–76, 2003.

[9] V. Brattka and K. Weihrauch. Computability on subsets of Euclidean space I: closed and compact subsets. *Theoretical Computer Science*, 219(1):65–93, 1999.

[10] M. Carl. *Ordinal Computability. An Introduction to Infinitary Machines.* De Gruyter Series in Logic and Its Applications. De Gruyter, 2019.

[11] M. Carl, L. Galeotti, and B. Löwe. The Bolzano-Weierstrass theorem in generalised analysis. 44(4):1081–1109, 2018.

[12] M. Carl, L. Galeotti, and R. Passmann. Realisability for infinitary intuitionistic set theory, 2020. Preprint arXiv:2009.12172v2.

[13] J. H. Conway. *On Numbers and Games.* A K Peters & CRC Press, 2000.

[14] S. Coskey and P. Schlicht. *Fundamenta Mathematicae*, 232(3):227–248.

[15] K.J. Devlin. *Constructibility.* Perspectives in Logic. Cambridge University Press, 1st edition, 2017.

[16] L. Galeotti. Computable analysis over the generalized Baire space. Master's thesis, ILLC Master of Logic Thesis Series MoL-2015-13, Universiteit van Amsterdam, 2015.

[17] L. Galeotti. *The theory of generalised real numbers and other topics in logic.* PhD thesis, Universität Hamburg, 2019.

[18] L. Galeotti, A. Hanafi, and B. Löwe. Relations between notions of gaplessness in non-Archimedean fields. *Houston Journal of Mathematics*, 46(4):1017–1031, 2020.

[19] L. Galeotti and H. Nobrega. Towards computable analysis on the generalised real line. In Jarkko Kari, Florin Manea, and Ion Petre, editors, *Unveiling Dynamics and Complexity : 13th Conference on Computability in Europe, CiE 2017, Turku, Finland, June 12–16*, volume 10307 of *Lecture Notes in Computer Science*, pages 246–257. Springer, 2017.

[20] G. Gherardi and A. Marcone. How incomputable is the separable hahn-banach theorem? *Notre Dame Journal of Formal Logic*, 221(4):393–425, 2008.

[21] H. Gonshor. *An Introduction to the Theory of Surreal Numbers*, volume 110 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 1986.

[22] J. D. Hamkins and A. Lewis. Infinite time Turing machines. *Journal of Symbolic Logic*, 65(2):567–604, 2000.

[23] T. Jech. *Set Theory: The Third Millennium Edition, revised and expanded.* Springer Monographs in Mathematics. Springer, 2006.

[24] Y. Khomskii, G. Laguzzi, B. Löwe, and I. Sharankou. Questions on generalised Baire spaces. *Mathematical Logic Quarterly*, 62(4-5):439–456, 2016.

[25] P. Koepke. Turing computations on ordinals. *Bulletin of Symbolic Logic*, 11(3):377–397, 2005.

[26] P. Koepke and B. Seyfferth. Ordinal machines and admissible recursion theory. *Annals of Pure and Applied Logic*, 160(3):310–318, 2009.

[27] E. Lewis. Computation with infinite programs. Master's thesis, ILLC Master of Logic Thesis Series MoL-2018-14, Universiteit van Amsterdam, 2018.

[28] L. Motto Ros. The descriptive set-theoretical complexity of the embeddability relation on models of large size. *Annals of Pure and Applied Logic*, 164(12):1454–1492, 2013.

[29] J.R. Munkres. *Topology.* Featured Titles for Topology Series. Prentice Hall, 2000.

[30] H. Nobrega. *Games for Functions: Baire Classes, Weihrauch Degrees, Transfinite Computations, and Ranks.* PhD thesis, Universiteit van Amsterdam, 2018.

[31] P. Odifreddi. *Classical Recursion Theory: The Theory of Functions and Sets of Natural Numbers.* Studies in Logic and the Foundations of Mathematics. Elsevier Science, 1992.

[32] B. Rin. The computational strengths of α-tape infinite time Turing machines. *Annals of Pure and Applied Logic*, 165(9):1501–1511, 2014.

[33] H. Rogers Jr. *Theory of recursive functions and effective computability.* MIT press, 1987.

[34] J. G. Rosenstein. *Linear Orderings*, volume 98 of *Pure and Applied Mathematics.* Academic Press, 1982.

[35] W. Rudin. *Principles of Mathematical Analysis.* International series in pure and applied mathematics. McGraw-Hill, 1976.

[36] R. Sikorski. On an ordered algebraic field. *Comptes Rendus des Séances de la Classe III, Sciences Mathématiques et Physiques. La Société des Sciences et des Lettres de Varsovie*, 41:69–96, 1948.

[37] R. Sikorski. Remarks on some topological spaces of high power. *Fundamenta Mathematicae*, 37(1):125–136, 1950.

[38] S. G. Simpson. *Subsystems of Second Order Arithmetic.* Perspectives in Logic. Cambridge University Press, 2nd edition, 2009.

[39] K. Weihrauch. *The Degrees of Discontinuity of some Translators Between Representations of the Real Numbers.* Number 129 in Informatik-Berichte. FernUniversität in Hagen, 1992.

[40] K. Weihrauch. *The TTE-Interpretation of Three Hierarchies of Omniscience Principles.* Number 130 in Informatik-Berichte. FernUniversität in Hagen, 1992.

[41] K. Weihrauch. *Computable Analysis: An Introduction.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2012.

[42] S. Willard. *General topology.* Courier Corporation, 2012.