Aspects of Computable Analysis

by

Michelle Helen Porter

A thesis

submitted to the Victoria University of Wellington

in fulfilment of the requirements for the degree of

Master of Science in Mathematics.

Victoria University of Wellington

2016

Abstract

Computable analysis has been well studied ever since Turing famously formalised the computable reals and computable real-valued function in 1936. However, analysis is a broad subject, and there still exist areas that have yet to be explored. For instance, Sierpiński proved that every real-valued function $f : \mathbb{R} \to \mathbb{R}$ is the limit of a sequence of Darboux functions. This is an intriguing result, and the complexity of these sequences has been largely unstudied. Similarly, the Blaschke Selection Theorem, closely related to the Bolzano-Weierstrass Theorem, has great practical importance, but has not been considered from a computability theoretic perspective. The two main contributions of this thesis are: to provide some new, simple proofs of fundamental classical results (highlighting the role of Π_1^0 classes), and to use tools from effective topology to analyse the Darboux property, particularly a result by Sierpiński, and the Blaschke Selection Theorem. This thesis focuses on classical computable analysis. It does not make use of effective measure theory.

Acknowledgements

Contents

Introduction

In the late 18$^{\text{th}}$ century, mathematical logic was facing something of a crisis. Early attempts to clarify the foundations of mathematics were resulting in inconsistencies and contradictions. In a bid to solve this problem, David Hilbert conjectured in 1900 that mathematics was complete; he believed that every question in the language of number theory should be decidable. It was in the early 1930s that the concept of 'by finite means' arose and Austrian Kurt Gödel disproved this conjecture [26]. An intuitive understanding of computability actually existed well before the 1930s, however, the proofs given during this time were mostly constructive. Phrases such as 'by finite means' or 'by constructive measures' were relatively standard, but lacked any precise definition. It was Alan Turing who famously formalised these concepts in his 1936 paper 'On Computable Numbers, with an Application to the Entscheidungsproblem'[60]. Turing defined a primitive machine, now known as the Turing machine, and used it not only to solve the Entscheidungsproblem, but also to define the computable reals. Turing called a real x computable if arbitrarily precise approximations of x could be computed by a machine. Or in other words, if there existed a computable Cauchy sequence of rationals with limit x.

His paper was influential for many reasons, particularly because before this time the foundations of computability were built upon the natural numbers (or finite strings), known as Type I objects. These objects are finitely describable, and therefore straightforward to work with. Real numbers, on the other hand are infinite objects, and so are not nearly as

easy to conceptualise. Because real numbers form the basis of analysis, by providing a neat and natural definition of the computable reals, Turing laid the foundation for a new branch of mathematics, known today as computable analysis.

Computable analysis would be extended and explored by many mathematicians in the following years, notably G. Ceitin [16], O. Demuth [22], R. Goodstein [27], S. Kleene [37], [38], G. Kreisel, D. Lacombe and J. Shoenfield [40], B. Kushner [43], A. Markov [49], [50], V. Orevkov [52], H. Rice [55], E. Specker [58], [59] and I. Zaslavsky [66],[67].[1] By around 1975, the development of computable analysis was largely complete. Texts summarising the area, including those by M. Pour-El and J. Richards [53] and O. Aberth [1] and [2], emerged.

In this thesis, we take a fresh look at some of the original results of computable analysis. We aim to provide a new take on some of those early proofs, highlighting the role of Π^0_1 classes. As well as this, we focus on two important results in classical analysis: a property closely tied to the Intermediate value Theorem, known as the Darboux property, and a generalisation of the Bolzano-Weierstrass Theorem, referred to as the Blaschke Selection Theorem.

A function f has the Darboux property on an interval if, for every a and b in this interval where $a < b$, and every y between $f(a)$ and $f(b)$, there exists an x in $[a, b]$ such that $f(x) = y$ [19]. Unfortunately, once it has been established that every computable real-valued function is continuous, the Darboux property on its own becomes less interesting. What is interesting is a result by Sierpiński that states that *every* real-valued function $f : \mathbb{R} \to \mathbb{R}$ is the pointwise limit of a sequence of Darboux functions [56]. This result is unusual and surprising, and we dedicate a large portion of this thesis to discussing how difficult it is to compute such a sequence of functions. While the Intermediate value Theorem has been analysed, see for example Pour-El and Richards [53], and Aberth [1], the

---

[1]This list is a sample, and by no means exhaustive. For further contributions, see reference list.

Darboux property, particularly the Sierpiński result, has not before been considered in this context.

The Blaschke Selection Theorem asserts that every infinite collection of closed, convex subsets in a bounded portion of $\mathbb{R}^n$ contains an infinite subsequence that converges to a closed, convex, nonempty subset of this bounded portion of $\mathbb{R}^n$ [5]. The Blaschke Selection Theorem is significant because it is related to one of the central theorems of classical analysis; that every bounded sequence of points in $\mathbb{R}^n$ has a convergent subsequence [7]. Largely unstudied from a computability theoretic perspective, in this thesis we explore how difficult it is to find Blaschke's convergent subsequences. We are also interested in how difficult it is to determine if a set is not convex.

We now give a brief outline of this thesis. The first chapter lists some prerequisites and notation that is relevant. Some knowledge of basic computability theory is assumed.

Chapter 2 introduces the relevant computable objects. The first section covers the computable reals while the second covers the computable real-valued function. We discuss some of the different definitions that are available to us, including Markov, Borel and Type II computability, and justify our choices. Chapter 3 introduces a computable subset of $\mathbb{R}^n$, utilising a particular type of distance function. We give a proof that the graph of a Type II computable function is computable on any interval, while the graph of a Markov computable function is upper semi-computable, but not necessarily computable, on any given interval.

Chapter 4 is dedicated to the Darboux property, specifically the Sierpiński result. We consider how complex particular Darboux functions are, and the consequences this has on the complexity of the approximating Darboux sequences. Complexity is discussed in terms of effective Baire classes 1 and 2. We show that any Baire class 2 function is the limit of a sequence of Baire class 2 Darboux functions. We are also interested in the effect that restricting the domain and range of the function has on the complexity of the approximating Darboux sequences. So we ex-

plore the Darboux property defined only on rational-valued functions. It turns out that any computable rational function is a limit of a sequence of computable rational Darboux functions. The last section of Chapter 4 briefly looks at the consequences of requiring a sequence of Darboux function to uniformly, rather than pointwise, approximate a function. The Bruckner, Ceder and Weiss paper [14] mostly inspire this section.

Chapter 5 introduces the singular point, and explores how hard it is to find these points for polynomials with computable real coefficients. This section has some connection with the Darboux property, but was mostly included for interest. Lastly, Chapter 6 is dedicated to the Blaschke Selection Theorem. We begin by giving a proof of the the Theorem, and then analyse this proof to show that $0''$ is sufficient to find a convergent subsequence of any appropriate collection of closed convex sets, and to compute its limit. We also prove that $0'$ is insufficient in the place of $0''$. Lastly, the final section of the chapter briefly looks into the complexity of convexity. We prove that $0'$ is not sufficient to deicide convexity in $\mathbb{R}^n$, but that the set of indices of closed convex sets is co-computably enumerable over $0'$.

At the beginning of each chapter or section we will clearly identify all original results.

Chapter 1

Prerequisites and Notation

We assume that the reader has some background in computability theory, but for those who are less familiar, we state some relevant definitions and results. For a brief introduction to computability, see [23].

Let $\varphi_1, \varphi_2, \ldots$ be a standard enumeration of the partial computable functions.

We will use $\mathbb{N}$ and ω interchangeably. *Cantor space* is the collection of infinite binary sequences, 2^ω. *Baire space* is the collection of infinite ω sequences, ω^ω. The most important difference between these two spaces is that Cantor space is compact, while Baire space is not.

We call natural numbers, or equivalently finite (binary) strings, Type I objects. We call real numbers, or equivalently infinite (binary) strings, Type II objects.[1] In general, Type n objects are sets of Type $(n-1)$ objects.

We define the notations Π_1^0, Σ_1^0 and Δ_1^0 as follows. A set $A \subseteq \mathbb{N}$ is Π_1^0 if there is a computable relation $R(x, y)$ such that $y \in A$ if and only if

$$\forall x R(x, y).$$

---

[1]Both equivalences follow by well-known isomorphisms.

A set A is Σ_1^0 if there is a computable relation $R(x, y)$ such that $y \in A$ if and only if

$$\exists x R(x, y).$$

A set A is Δ_1^0 is it is both Π_1^0 and Σ_1^0.

We note that a set A is computably enumerable (c.e.) if and only if A is Σ_1^0.

One of the difficulties of computable analysis is dealing with Type II, rather than Type I, objects. Type I objects can be expressed finitely, and therefore, collections of Type I objects form *sets*. Type II objects, on the other hand, are infinite, and therefore, collections of Type II objects form *classes*. For this reason we also need to define Π_1^0, Σ_1^0 *classes*. We do this now.

A *tree* is a subset of $2^{<\omega}$ that is closed under initial segments. We call an infinite sequence $P \in 2^\omega$ a *path* through a tree T if for all $\sigma \prec P$ we have $\sigma \in P$. The collection of all paths in T is denoted $[T]$.

For every string $\sigma \in 2^{<\omega}$ (the collection of finite strings) we define a *basic open class* to be

$$[\![\sigma]\!] = \{x : x \in 2^\omega \text{ and } \sigma \prec x\}.$$

The *open classes* of Cantor space are unions of basic open classes. A class $\mathcal{A} \subseteq 2^\omega$ is *effectively open* if $\mathcal{A} = [\![A]\!]$ for some computable set $A \subset 2^{<\omega}$. A class $\mathcal{A}$ is Σ_1^0 if there is a computable relation R such that

$$\mathcal{A} = \{x : \exists n R(x \upharpoonright n)\}.$$

A set $\mathcal{A}$ is *effectively open* if and only if $\mathcal{A}$ is Σ_1^0.

A class $\mathcal{C} \subseteq 2^\omega$ is Π_1^0 if there is a computable relation R such that

$$\mathcal{C} = \{x : \forall n R(x \upharpoonright n)\}.$$

Or equivalently, a subset of 2^ω is a Π_1^0 class if it is equal to $[T]$ for

some computable tree T. An example of a Π_1^0 class is any collection of separating sets $\{X : A \subseteq X \text{ and } X \cap B = \emptyset\}$, where A and B are disjoint c.e. sets.

$\mathcal{C}$ is *effectively closed* if and only if $\mathcal{C}$ is Π_1^0. A class $\mathcal{C}$ is *closed* if its complement is open. A class $\mathcal{C}$ is effectively closed if its complement is effectively open.

We define a *computable metric space* to be a separable, complete metric space (Polish metric space) $\mathcal{X} = (X, d, Y)$ with metric d and countable dense subset Y such that, given ϵ and $x, y \in Y$ in our space, there exists an algorithm that computes $d(x, y)$ to within ϵ. That is, there exists a computable function $f(x, y, \epsilon)$ that outputs a value to within ϵ of $d(x, y)$.

We call a point x in a metric space M an *accumulation point* of $A \subset M$ if every neighbourhood of x has a point in A other than x.

Note that a *neighbourhood of x* is simply any set that contains an open set that contains x. That is, N is a neighbourhood of x if there exists an open set O such that $x \in O \subseteq N$.

Let $X \subset \mathbb{R}^n$. Then $\overline{X}$ is the closure of X.

Unless otherwise mentioned, we consider $B(x, \epsilon)$ to be the open ball in the appropriate space, with center x and radius ϵ.

For two sets $X, A \subset \mathbb{R}^n$, let $X \backslash A = \{x : x \in X \text{ and } x \notin A\}$.

Lastly, as convergent functions will play a significant role in this thesis, we emphasise the distinction between pointwise and uniform convergence.

A sequence of functions $f_1, f_2, \ldots$ (each sharing the same domain and co-domain) is said to *converge pointwise* to a function f if and only if

$$(\forall x \in \text{ dom}(f)) \lim_{n \to \infty} f_n(x) = f(x).$$

Pointwise convergence is probably the most natural way to define con-

vergence, but it is not always as well behaved as you might expect. It does not need to preserve, for example, boundedness, continuity or differentiability.

For example, consider the sequence of functions $f_n : [0, 1] \to \mathbb{R}$ defined by $f_n(x) = x^n$. This sequence converges pointwise to the following function.

$$f(x) = \begin{cases} 1 & \text{if } x = 1 \\ 0 & \text{otherwise} \end{cases}$$

Notice that, while each function f_n is continuous on $[0, 1]$, their pointwise limit f is not.

Uniform convergence is a stronger condition and forces much better behaviour. For instance, it preserves continuity.

A sequence of functions $f_1, f_2, \ldots$ (each sharing the same domain and co-domain) is said to *uniformly converge* to a function f if, for all $\epsilon > 0$ there exists N such that $n > N$ implies

$$(\forall x \in \text{dom}(f)) \; |f(x) - f_n(x)| < \epsilon.$$

The important point here is that N depends only on ϵ and not on x. In a pointwise convergent sequence, N may depend on both ϵ *and* x. The example we gave above does not uniformly converge; when $0 \leq x < 1$ and $0 < \epsilon < 1$ we have $|f_n(x) - f(x)| = x^n < \epsilon$ if and only if $0 \leq x < \epsilon^{\frac{1}{n}}$. But $\epsilon^{\frac{1}{n}} < 1$ for all n. And so, for all N there exists a y such that $\epsilon^{\frac{1}{N}} < y < 1$, therefore $|f_m(y) - f(y)| = y^m > \epsilon$, for $m > N$. That is, N must depend on both ϵ and x.

Chapter 2

First Attempts and Definitions

In this chapter we will shed some historical light on the definitions used in this thesis. We begin with the computable real numbers. We give a new direct proof of Theorem 2.2.3, a result due to Kreisel, Lacombe and Shoenfield [40]. We also construct an original Markov computable function that cannot be extended to any continuous function on $\mathbb{R}$ (Example 2.2.7).

2.1 Defining a computable real

The concept of a real number has existed for centuries, but was formalised only around 150 years ago. In the early 19th century the French mathematician Augustin-Louis Cauchy defined a Cauchy sequence to be a sequence of rationals $x_1, x_2, \ldots$ such that, for all $\epsilon > 0$ there exists an n such that, if $m > n$, then $|x_n - x_m| < \epsilon$. It was then in 1871 that Georg Cantor took this construction and used it to formalise the notion of the real number.[1] Cantor defined a number x to be real if it was the limit

---

[1]In this year Richard Dedekind also developed Dedekind cuts, an equivalent definition.

of a Cauchy sequence of rationals. It was in effectivising this definition that computable analysis was born.

Armed with this definition, it would now seem natural to us that a computable real should involve a Cauchy sequence converging in some algorithmic manner. Unfortunately, at this time there existed no formal notion of computation. However, at the turn of the 20th century there was certainly an intuitive sense of what an algorithmic method was. For example, see the works of Dehn [20], Hermann [32], Kronecker [41], and von Mises [63].[2]

One notable paper that demonstrated this was by Borel in 1912 (coincidentally the year of Alan Turing's birth). In his paper Borel claims that a real x is 'computable' if, given any natural number n, we can obtain a rational q within $\frac{1}{n}$ of x [10].[3] What Borel means by 'computable' is uncertain, particularly since it would be another 20 years before any formal notion of computation emerged. We also hesitate to speculate what Borel intended when he spoke of 'obtaining' a rational close to x. However, in a footnote Borel writes;

> I intentionally leave aside the practical length of operations, which can be shorter or longer; the essential point is that each operation can be executed in finite time with a safe method that is unambiguous.

While some students of history disagree about Borel's intention, if our understanding is correct, his intuition, at least, seems reasonable; a real should is computable if we can, in finite time, give an approximation of it with arbitrary accuracy.

It was not until 1936 that Turing tackled the definition in his paper 'On computable numbers' [60]. While Church, Kleene and Post were

---

[2]For English translation of [20], see [21]. For an English translation of [32], see [33]. For an English translation of [41], see [42]. For an English translation of [63], see [64].

[3]Quotes and comments from Borel's paper [10] are based on a translation (French to English) by Avigad and Brattka [3].

all looking into this area around this time, Turing's paper is accepted as the most intuitively clear.[4] Turing begins by stating that a real x is computable if its decimal expansion can be output by finite means.[5] He then goes on to define the computing machine, and that 'by finite means' refers to a machine that can output a sequence of symbols given some fixed amount of information.

Noting that Turing called a machine that writes only a finite number of symbols *circular*, he finally states;

> A sequence is computable if a circular-free machine can compute it. A number is computable if it differs by an integer from the number calculated by a circular-free machine.

Simply put, a real x is considered computable if there exists a Turing machine that, given no input, outputs a binary decimal expansion of x.

There is, however, a problem with this definition, as Turing later noted in his correction [61]. He believed that if we can compute a rational q_i for all i such that $|x - q_i| < 2^{-i}$ (which he called the 'intuitive requirement'), then x should also be considered computable in the context of his original definition, and vice versa.

We immediately have one direction; if we have a binary expansion of a real x, then the truncated binary expansion will provide a sufficient rational in the sense of the second definition. It is in the other direction that Turing noted a disparity. For example, suppose we have a sequence of rational numbers $(q_i)_i$ that approach x as above, and $q_1 = \frac{1}{2}, q_2 = \frac{1}{2}, q_3 = \frac{1}{2}, \ldots$. Then we have no way of knowing what the first binary point of x should be, because our sequence may move above or below $\frac{1}{2}$ at any stage.

To correct this non-uniformity, Turing goes on to modify the way he associates computable numbers with computable sequences. His solution

---

[4]See any historical discussion about Turing, for example, [23]

[5]It is quite remarkable that Turing based his notion of computability on a subset of $\mathbb{R}$ rather than the integers.

is a formula that incorporate both concepts. The details, which we do not give here, can be found in [61]. Being a rather cumbersome way to consider a real number, this thesis will bypass any separation of the computable reals and computable sequence definitions. Instead, we will consider a real x to be computable if we can computably produce a rational approximation as close to x as we would like. As a consequence, every rational is computable, where we output a rational in some appropriate way, for example by giving its Gödel number.

And so we finally give the definition that we will be using;

Definition 2.1.1. A *Cauchy name* for a real x is a sequence $(x_i)_i$ of rationals that converge rapidly to x. That is, for every k and $j \geq k$, $|x_j - x| < 2^{-k}$.

Definition 2.1.2. A real number x is *computable* if it has a computable (rapidly converging) Cauchy name.

We call the collection of all computable real numbers $\mathbb{R}_c$. In general, if x is a computable real we will write a computable Cauchy name as $(x_i)_i$.

2.1.1   Preliminary results and computable reals

Before moving on, we take a moment to give some initial thoughts about the computable reals.

We first note that the definition we have given of a computable real could be replaced with a number of equivalent definitions. For example, we could have called a real x is computable if there exists a computable sequence of shrinking intervals uniquely enclosing x. That is, there exist two computable rational sequences $(u_i)_i$ and $(v_i)_i$ such that $u_1 < u_2 < \ldots < u_n < \ldots x \ldots < v_n < \ldots < v_2 < v_1$. Brattka, Hertling and Weihrauch give this and some other alternate definitions in [11]. However, the one we have chosen is the most intuitive and appropriate in the context of this thesis.

Irrespective of the chosen definition, it very quickly becomes apparent that a computable real is not as nice to work with as we would perhaps like. Deciding whether two real numbers are the same is a natural and seemingly simple question. Maybe we would not expect to be able to decide this for *any* two real numbers, but perhaps for at least two *computable* reals. It is one of the great tragedies of computable analysis that this is not the case.

The difficulty arises because we are dealing with Type II objects. It is easy to decide whether two natural or rational numbers are equal, however, while two Cauchy names may seem to be very close for a long time, always at some later stage we may observe divergence. Consequently, if two computable reals a and b are not equal, we will see at some point that $a < b$ or $b < a$. However, if $a = b$, this can never effectively be concluded.

Theorem 2.1.3. *(Folklore, implicit in Turing [60]) The following operations cannot be computably decided: $x = y$, $x \leq y$.*

Proof: Suppose to the contrary. Fix e. Define a computable sequence of rationals $(x_i)_i$ such that x_i represents the state of the eth machine at stage i on input e. That is, $x_0 = 0$ and we let $x_i = x_{i-1} + 0.\overbrace{00\ldots00}^{i-1}1$ if $\varphi_e(e)[i] \downarrow$ and $x_i = x_{i-1} + 0.\overbrace{00\ldots00}^{i-1}0$ otherwise. $(x_i)_i$ is a computable sequence of rationals, and we let x be the limit to this sequence. We then ask whether $x = 0$? If yes, then we know $\varphi_e(e)$ does not halt, and if no it does. Contradiction. $x \leq y$ follows similarly. $\square$

Notice that the complexity of equality is at worst $0'$; to decide if $x = y$, given respective Cauchy names $(x_i)_i$ and $(y_i)_i$, simply ask whether $(\forall n)|x_n - y_n| < 2^{-(n-1)}$.

Theorem 2.1.4. *(Folklore) The following operations are computable: $x + y$, $x - y$, xy, if $y \neq 0$ then $x \div y$, if $x > 0$ then $\exp(x)$ and $\log(x)$, $\sin(x)$, $\cos(x)$, $\tan^{-1}(x)$, $\max(x, y)$, $\min(x, y)$ and $\sqrt{x}$ as long as $x \geq 0$.*

In each case, given the Cauchy names of x and y, apply an appropriate uniform procedure to each Type II object. For example $x_1 + y_1, x_2 + y_2 \ldots$ is the computable Cauchy name of $x + y$. Later in the thesis we will (often implicitly) use these computable operations.

Lastly, we list a few facts to keep in mind:

1. Because there are only countably many computable functions, there are countably many computable reals and hence uncountably many noncomputable reals.

2. $\mathbb{R}_c$ is dense in $\mathbb{R}$ and forms a real closed field [55].

3. It is not enough that a real has a Cauchy name, this sequence must be *computable* for a real x to be considered computable.

4. There are infinitely many distinct Cauchy names for any computable real, and no computable listing of every computable Cauchy name. If there were, we could diagonalize and arrive at a contradiction.

5. While we cannot list every computable Cauchy name, we can build a computable tree in Baire space whose paths represent Cauchy names. We represent the rationals with Gödel numbers, and the nth element in a branch represents the nth term in a potential Cauchy sequence. We 'kill' a branch at stage/height n if the next element to be added is further than 2^{-m} away from element m for all $m < n$. Note that, while the tree may be computable, the paths need not be. We will use the fact that the Cauchy names form a Π_1^0 class later on.

2.1.2   Computable metric space and computable sequence

Recall that we defined a *computable metric space* to be a separable complete metric space (polish metric space) $\mathcal{X} = (X, d, Y)$ with metric d and

countable dense subset Y such that, given ϵ and $x, y \in Y$ in our space, there exists an algorithm that computes $d(x, y)$ to within ϵ. Notice that $(\mathbb{R}, d, \mathbb{R}_c)$ is a computable metric space; in the separable complete metric space $(\mathbb{R}, d)$, with the usual metric d, $\mathbb{R}_c$ is a countable dense subset, and by Theorem 2.1.4, for any $x, y \in \mathbb{R}_c$ we can compute $d(x, y) = |x - y|$ to within ϵ. This is the space we will usually be working in, however, we note in passing that the concept of a 'computable real' can be generalised to other computable metric spaces. Consider the following example.

Example 2.1.5. Let X be the collection of all real-valued functions $f :$ $[0, 1] \to \mathbb{R}$. Define a metric $d_X(f, g) = \sup\{|g(x) - f(x)| : x \in [0, 1], f, g \in X\}$. Then the collection Y of all polynomials $p : [0, 1] \to \mathbb{R}$ is a countable dense subset of X. For any two polynomials p_1, p_1, we can compute $d_X(p_1, p_2)$ to within ϵ, hence (X, d_X, Y) is a computable metric space. We can then define a function $f \in X$ to be 'computable' in this space if it is the limit of a fast converging sequence of polynomials in Y.

Note that many of the results to follow can be generalised to computable metric spaces.

Lastly, we define a computable sequence of real numbers.

Definition 2.1.6. We call a sequence of real numbers $(r_i)_i$ *computable* if there exists a computable sequence of rationals $(q_{i,k})_{i,k}$ such that for all i, $(q_{i,k})_k$ is a computable Cauchy name for r_i.

2.2 Defining a computable real-valued function

Defining the computable real-valued function again highlights the difficulties of working with Type II, rather than Type I, objects. Computable Type I functions simply take finitely describable objects to other finitely describable objects. How do we extend this concept to the infinitely describable reals?

One reasonable approach could be to consider a function f computable if there exists an algorithm that, by observing the first n terms of a Cauchy name for x, outputs the first n terms of the Cauchy name of $f(x)$. However, if we used this definition, every computable function would necessarily be Lipschitz.[6] Unfortunately this would exclude some functions that we would like to consider computable, for instance, $f(x) = x^2$.[7] These kinds of functions have been studied, and while they have relevance in computable randomness (for example see [23]), we would like to give our computable functions more freedom. Ideally, we would still like some finite number of terms of a Cauchy name for x to provide sufficient information to compute the nth term in the Cauchy name of $f(x)$. However, we do not want to put restrictions on how large this number may be.

A natural place to start is by applying this concept to functions with domains restricted to the computable reals. Turing also started here, in 1936.

2.2.1   A computable function $f : \mathbb{R}_c \to \mathbb{R}_c$

Defining the computable real and computable real-valued function is a similar process. So it does not come as a surprise that before the 1930s there existed an intuitive understanding of how a computable real-valued function should behave, even if there did not exist a widely accepted formal classification. For example, in the same paper that Borel wrote about his 'computable real', he also provided some insight into how he viewed a computable real-valued function [10]. Borel believed that a function f should be called computable if, given a 'computable number'

---

[6]A function f is Lipschitz if for all $a, b \in dom(f)$, $|f(a) - f(b)| \leq L|a - b|$ for some constant L. For more information see [31].

[7]Consider a real x and Cauchy name with initial term $\frac{3}{2}$. The real x could be anything in the interval $[1, 2]$, which means the image of x under f could be anything in the interval $[1, 4]$. There is no rational we can output as the first term in a new Cauchy name that could satisfy all values in $[1, 4]$. Therefore, we need more information about the Cauchy name of x before we can give the first term in the Cauchy name of $f(x)$ (for example the second term should be sufficient).

α, you can compute $f(\alpha)$ to within $\frac{1}{n}$, for any n.

Borel does not specify what he intends by 'method', nor is he clear whether he expects a computable function to be that which takes a *method* to compute a real to another *method* to compute a real, or instead, an *approximation* to an *approximation*. Regardless of his motivation, he seems to give a reasonable suggestion for a computable real-valued function. Indeed, he goes on to assert that a function cannot be computable unless it is continuous at every computable value of a given variable, which will become a necessary property in every definition we will go on to discuss.

Of course, it was Turing who would formalise a widely accepted notion of the computable real-valued function. Like Borel, Turing initially restricted his focus to only those functions defined on $\mathbb{R}_c$. He declared that computable functions cannot be defined on all real values because there is no general way to describe all real numbers. We will soon see that this is not the case, and in fact, Turing's own position would change in later years when he formalised the oracle Turing machine. However, for now, we will continue to follow his original approach.

If x is a computable real, then by definition, there must exist a primitive machine φ_e that corresponds to a Cauchy name for x. Logically extending this idea, Turing defined a function $f : \mathbb{R}_c \to \mathbb{R}_c$ to be computable if there exists a total computable function ψ such that $\psi(e)$ is the index of a primitive machine that corresponds to the Cauchy name of $f(x)$. For a more formal classification, see Turing's original paper [60]. Turing's function essentially takes a *method* to compute x to a *method* to compute $f(x)$. The Russian school later adopted and further developed this idea, and today Turing's computable function is more commonly known as Markov computable.

Markov's version of a computable function $f : \mathbb{R}_c \to \mathbb{R}_c$ follows [49].

Let $\varphi_1, \varphi_2 \ldots$ be a standard enumeration of the partial computable functions. We call $e \in \mathbb{N}$ an *index name* of $x \in \mathbb{R}_c$ if e is the index of the

partial machine that computes a rational Cauchy name of x.

Definition 2.2.1. We call a function $f : \mathbb{R}_c \to \mathbb{R}_c$ *Markov computable* if there exists a partial computable function $\nu : \mathbb{N} \to \mathbb{N}$ such that, given any index name e of $x \in \mathbb{R}_c$, $v(e)$ exists, and is an index name of $f(x)$

We call the function $\nu : \mathbb{N} \to \mathbb{N}$ the *index function* of f.

It turns out that the Turing-Markov definition (which we will now refer to as 'Markov') is not the only interpretation of the computable function $f : \mathbb{R}_c \to \mathbb{R}_c$. *Borel* computability, which in contrast takes approximations (rather than methods) to approximations, is also a reasonable definition worth considering. In the introduction to this section, we mentioned the idea of using a finite number of bits of the Cauchy name of real x to give some approximation of $f(x)$. Borel computability formalises this notion.[8]

Definition 2.2.2. Let $(x_i)_i$ be a Cauchy name of x and $|q - f(x)| < 2^{-n}$. We call a function $f : \mathbb{R}_c \to \mathbb{R}_c$ *Borel computable* if there exists an oracle Turing Machine Φ such that, for all $x \in \mathbb{R}_c$ and $n \in \mathbb{N}$, $\Phi^{(x_i)_i}(n) = q$.

This definition is nice, especially in a modern context. Fortunately, we do not need to spend time agonising over which notion of computability is more appropriate, because it turns out that Markov and Borel computability are equivalent! This greatly simplifies things for us later on, as we will be able to use whichever notion is more contextually convenient.

The following theorem was originally proved by Kreisel, Lacombe and Shoenfield in 1959 [40], and later by Ceĭtin [16] in 1967. We give a new direct proof of this result that shows that Markov computable functions have a particular effective continuous property.

Theorem 2.2.3. *(Kreisel, Lacombe, Shoenfield [40]) A function f is Markov computable if and only if it is Borel computable.*

---

[8]This definition is not really attributed to Borel, however, as these types of functions are classically referred to as 'Borel computable' we will stick with this notation to avoid confusion.

Proof: ($\Leftarrow$) Let $f : \mathbb{R}_c \to \mathbb{R}_c$ be a Borel computable function and e the index name of a computable real x. The function f is Borel computable, so given the Cauchy name of x we have access to a Cauchy name $(f(x)_i)_i$ of $f(x)$.[9] Define a function $\nu : \mathbb{N} \to \mathbb{N}$ such that $\nu(e) = e'$. Then ν is a partial computable index function and so f is Markov computable.

($\Rightarrow$) This direction is not as straightforward. Initially, you may think to approach it in a similar manner to the backwards direction. However, if x is a computable real, but we do not have access to a *computable* Cauchy sequence converging to x, then there is no machine, and therefore no index, that outputs this particular approximation of x. This means we cannot find the index name of $f(x)$ using ν.

We will prove that a Markov computable function f is Borel computable in a few steps. First we show that if a Markov computable function f is effectively continuous on $\mathbb{R}_c$ then f is Borel computable. We then prove that there exists a unique c.e. set of open balls that ensures that every Markov computable function is effectively continuous. First, let us define what we mean by 'effectively continuous'.

Definition 2.2.4. Let $f : \mathbb{R}_c \to \mathbb{R}_c$ be a function defined on an interval $I \subseteq \mathbb{R}_c$. The function f is *effectively continuous* on I if there exists a computable function $d(\epsilon, a)$ such that, for all $x, a \in I$ and $\epsilon > 0$,

$$|x - a| < d(\epsilon, a) \implies |f(x) - f(a)| < \epsilon.$$

Claim 1: If a Markov computable function $f : \mathbb{R}_c \to \mathbb{R}_c$ is effectively continuous on $\mathbb{R}_c$ then f is Borel computable.

Proof of claim 1: Let f be an effectively continuous Markov computable

---

[9]Define a function $\phi(m, n) = f(x)_n$, where $f(x)_n$ is the nth term in $f(x)'s$ Cauchy name. By the S-m-n Theorem we can find a total computable function g such that $\phi(m, n) = \varphi_{g(m)}(n)$ for all $n \in \mathbb{N}$. By the Recursion Theorem g has a fixed point. That is, there exists an e' such that $\varphi_{g(e')}(n) = \varphi_{e'}(n)$. Then $\phi(m, n) = \phi(e', n) = \varphi_{g(e')}(n) = \varphi'_e(n)$, and so e' is the index of the Cauchy name of $f(x)$, and is computable from g.

function. To prove our claim, we must show that, for any computable real $x \in \mathbb{R}_c$ and $n \in \mathbb{N}$, we can compute a nth approximation, and therefore Cauchy name, of $f(x)$.

Fix an arbitrary computable real a with (possibly noncomputable) Cauchy name $(a_i)_i$. All we need to do is find $b \in \mathbb{R}_c$ such that $|f(b) - f(a)| < 2^{-(n+1)}$ and take the $(n+1)^{th}$ approximation of $f(b)$. If we call $f(b)_{n+1}$ the $(n+1)^{th}$ approximation of $f(b)$, then $f(b)_{n+1}$ will be within 2^{-n} of $f(a)$, and so will act as a sufficient approximation of $f(a)$. More precisely, $|f(b)_{n+1} - f(a)| \leq |f(b)_{n+1} - f(b)| + |f(b) - f(a)| < 2^{-(n+1)} + 2^{-(n+1)} = 2^{-n}$.

We can find such a $b \in \mathbb{R}_c$ as follows: Let m be least such that $d(2^{-(n+1)}, a) \geq 2^{-m}$. Wait for a $q \in \mathbb{Q}$ such that $|q - a_{m+1}| < 2^{-(m+1)}$ and consider the corresponding constant Cauchy name $q, q, q, q, \ldots$ with index name e. Then

$$
\begin{aligned}
|q - a| &\leq |q - q_{m+1}| + |q_{m+1} - a_{m+1}| + |a_{m+1} - a| \\
&< 0 + 2^{-(m+1)} + 2^{-(m+1)} \\
&= 2^{-m}
\end{aligned}
$$

Because $|q - a| < 2^{-m} \leq d(2^{-(n+1)}, a)$, we have that $|f(q) - f(a)| < 2^{-(n+1)}$. Setting $q = b$, we are done.[10] $\square$

We now need to prove that a Markov computable function is in fact effectively continuous on $\mathbb{R}_c$. We do this by showing that there exists a c.e. set W with the following properties.

Fix a Markov computable function $f : \mathbb{R}_c \to \mathbb{R}_c$ and let W be a c.e. set of pairs of open balls such that:

1. If the pair of open balls $(b_1, b_2) \in W$, then for all computable real

---

[10]In detail; we now take the partial function with index e and input it into the index function ν of f. Then $\nu(n)$ is the index of the partial function that outputs a Cauchy name for $f(b)$. We take the $(n+1)^{th}$ approximation of $f(b)$ and are done.

numbers x contained in b_1, $f(x)$ is contained in b_2.

2. For all $x \in \mathbb{R}_c$ and for all k, there exists a pair of open balls $(b_1, b_2) \in W$ such that: b_2 is a ball of any radius 2^{-k}, the center of b_1 is a term in a Cauchy name of x and the center of b_2 is a term in a Cauchy name of $f(x)$. More precisely, if e is an index name of $x \in \mathbb{R}_c$ and ν is the index function of f, then for all k there exists a k' such that $(B(x_{k'}, 2^{-k'}), B(f(x)_k, 2^{-k})) \in W$. The rational $f(x)_k$ is a computable k^{th} approximation of $f(x)$, whose index name is $\nu(e)$ (and similarly for x_k).

Then if such a set W exists, f must be effectively continuous on $\mathbb{R}_c$. This fact follows directly from the definition of W. However, the details are given below for completeness.

Suppose we had such a set W with Properties 1 and 2 above. We need to define a computable function $d(2^{-m}, x)$ for all $x \in \mathbb{R}_c$ and $m \in \mathbb{N}$ as given above. Fix $a \in \mathbb{R}_c$ (we are given an index name of a, and hence have access to a computable Cauchy name of a) and $n \in \mathbb{N}$. Let $a_1, a_2, \ldots$ and $f(a)_1, f(a)_2, \ldots$ be Cauchy names of a and $f(a)$ respectively.

To define $d(2^{-n}, a)$, we wait for a particular pair (b_1, b_2) to be enumerated into W. We know that there exist Cauchy names $a_1, a_2, \ldots$ and $f(a)_1, f(a)_2, \ldots$ of a and $f(a)$ respectively such that $(b_1, b_2) \in W$ where $b_2 = B(f(a)_{n+3}, 2^{-(n+3)})$ and $b_1 = B(a_k, 2^{-k})$ for some k by Property 2 of W.

When we observe such a pair, set $d(2^{-n}, a) = 2^{-(k+1)}$.

Justification: if $d(2^{-n}, a) = 2^{-(k+1)}$ then $|x - a| < d(2^{-n}, a)$ if and only if $x \in B(a, 2^{-(k+1)})$. But by Property 1 of W, if $x \in b_1 = B(a_k, 2^{-k}) \cap \mathbb{R}_c$ then $f(x) \in b_2 = B(f(a)_n + 3, 2^{-(n+3)}) \subset B(f(a), 2^{-n})$. Hence, if $|x-a| < d(2^{-n}, a)$ then $|f(x) - f(a)| < 2^{-n}$, and so f is effectively continuous.

Finally, we need only show W exists.

Claim 2: We can enumerate a c.e. set W as described above for any

Markov computable function f.

Proof of claim 2: Note that the Recursion Theorem is used implicitly during this proof (and thesis). In general we omit these details for simplicity. Let $e_1, e_2, \ldots$ be a computable listing of indices corresponding to all possibly partial Cauchy sequences. That is, for any index i, the machine with index e_i copies the sequence output by φ_i until a stage where the sequence no longer looks Cauchy. If we observe a machine e_i outputting some rational that is too far away from the previously observed term in the sequence, we halt e_i on the last viable output forever. So every index in our list must correspond to a sequence that *looks* Cauchy at every stage, although some may be partial.

We now refine this list. Let $\nu : \mathbb{N} \to \mathbb{N}$ be the index function of Markov computable function f. Recall that if e is the index of a partial machine that outputs a Cauchy name for real x, then $\nu(e)$ is the index of a partial machine that outputs a Cauchy name for real $f(x)$. Build a new list of indices as follows; let $e(i, k)$ represent the index of the machine that copies machine e_i on all outputs until a stage s is reached where the machine with index $\nu(e(i,k))$ halts and outputs a viable k^{th} approximation to some rational (hopefully the image under f of whatever e_i represents). More precisely, copy φ_{e_i} on all outputs until we reach a stage s where $\varphi_{\nu(e(i,k))}(k)[s] = r_k \in \mathbb{Q}$, and in preceding stages has output $r_1, \ldots, r_{k-1}$, where $(\forall j)(\forall i) j < i < k, |r_i - r_j| < 2^{-j}$. If this is observed, machine $e(i, k)$ pauses. This machine will restart at a later stage if W does not meet the conditions given. Recall that, by stage s machine $e(i, k)$ has output some finite sequence of rationals $q_1, \ldots, q_{k'}$ that appear to be Cauchy. We now enumerate the ball $(B(q_{k'}, 2^{-k'}), B(r_k, 2^{-k}))$ into W.

We claim that W is as defined above. Suppose that Property 1 does not hold. That is, there exists a pair $(b_1, b_2) = (B(q_{k'}, 2^{-k'}), B(r_k, 2^{-k})) \in W$ such that there is some $a \in b_1 \cap \mathbb{R}_c$, yet $f(a) \notin b_2$. At some point we will enumerate a pair (b_3, b_4) into W such that $b_3 \in b_1$ but $b_2 \cap b_4 = \emptyset$ (because a is a computable real). Suppose this occurs at stage s, and let $e(i, k)$ and $e(j, l)$ be the indices of machines that were responsible for

(b_1, b_2) and (b_3, b_4)'s enumeration into W respectively. Let φ_{e_j} be total, and output a Cauchy name of a. This means $e(j, l)$ will be copying a true Cauchy name (see note on this later).

We now instruct the machine $e(i, k)$ to copy the Cauchy name currently being copied by machine $e(j, l)$. This is allowed because $b_3 \in b_1$, and so $e(i, k)$ will still output a viable Cauchy name. That is, suppose by this stage $e(j, l)$ has copied rationals $a_1, \ldots a_m$ in a Cauchy name of a. Assuming $m > k'$, we instruct $e(i, k)$ to copy the machine φ_{e_j} from this point onwards (if $m < k'$ just wait until we have seen the k'^{th} term in the Cauchy name of a and copy after that point). Then $r_1, r_2, \ldots r_{k'}, a_m, a_{m+1} \ldots$ is a Cauchy name of a and $\varphi_{\nu(e(i,k))}(k)[s] = r_k$ (by the Use Principle), but $f(a) \notin B(r_k, 2^{-k})$.[11] Since ν is an index function, we have a contradiction.

Note that we assume $e(j, l)$ is copying a true Cauchy name of a. We cannot, of course, computably know whether the machine φ_{e_j} is total, but we know that such a machine exists, and that is sufficient.

And so 1 holds. 2 follows easily; every computable real has a Cauchy name represented by some index in our list, and hence the desired balls must be enumerated into W at some stage.

Therefore the c.e. set W exists as claimed, and this proves both the claim and the result. $\square$

We will now use Borel and Markov computability interchangeably.

Finally, we give one last definition.

Definition 2.2.5. We call a function $f : \mathbb{R}_c \to \mathbb{R}_c$ *Banach-Mazur computable* (also known as *sequentially computable*) if f maps any given computable sequence $(r_i)_i$ of real numbers into a computable sequence

---

[11]The *use* of a converging oracle computation $\Phi^A(n)$ is $z + 1$ for the largest z such that $A(z)$ is queried during the computation. Let the use function be *Use*: $\mathbb{N} \to \mathbb{N}$. That is, $Use(\Phi^A(n)) = z + 1$ from above. The Use Principle is as follows; let Φ^A be a converging oracle computation and B a set such that $B \upharpoonright Use(\Phi^A(n)) = A \upharpoonright Use(\Phi^A(n))$. For more details see, for example, [23] Section 2.

of real numbers $(f(r_i))_i$.

Banach and Mazur developed this type of computability in the 1930s [4]. It does not seem very natural in our opinion and is known to be different to those given above. A function that is Markov computable must be Banach-Mazur computable, and while the converse holds in some cases, it is not true in general. See [35] for Hertling's construction of a Banach-Mazur computable but not Markov computable function on $\mathbb{R}_c$.[12] Banach-Mazur computability is too general for our purposes because it characterises functions as computable even if they may not be computed in the typical sense; by a Turing Machine. This type of computability is not widely studied and does not play a large role in this thesis.

Notice that all of these types of computability require the function in question to be continuous on its domain. This fact is unsurprising when you think about it, and will remain a requirement when we look at computable function defined on all of $\mathbb{R}$. However, we will soon see that not every Markov/Borel computable function can be extended to even a continuous function on $\mathbb{R}$, much less a computable one. And with this in mind, we move on to defining the computable real-valued function.

2.2.2   A computable function $f : \mathbb{R} \to \mathbb{R}$

We now reconsider the definition of a computable real-valued function. In the previous section, we restricted our attention to only the computable reals. However, as all real values are Type II objects, it could be argued that it is more natural to consider a computable process as taking one Type II object to another (rather than just those that happen to be computable). Kleene first investigated this notion in 1952 [37]. He considered a 'computable real-valued function' to involve an effective procedure that takes Type II objects to Type II objects on the whole space. Let us consider what that means.

---

[12]Hertling has also written other papers about Banach-Mazur computability, notably 'Banach-Mazur computable functions on metric Spaces' [34].

Given an effectively converging Cauchy sequence in Baire space, we would like to map this sequence is some uniform way to another effectively converging Cauchy sequence. We give Kleene's solution below.

Definition 2.2.6. Let $(x_i)_i$ be a Cauchy name of x and $|q - f(x)| < 2^{-n}$. We call a function $f : \mathbb{R} \to \mathbb{R}$ *Type II computable* if there exists an oracle Turing Machine Φ such that, for all $x \in \mathbb{R}$ and $n \in \mathbb{N}$, we have $\Phi^{(x_i)_i}(n) = q$.

Notice that this definition is simply an extension of Borel computability. The only difference is that we now allow x to take any real value, rather that restricting x to $\mathbb{R}_c$. We sometimes drop the 'Type II' and just call these functions computable.

There are a number of equivalent definitions scattered throughout the literature that we could have used in the place of Definition 2.2.6. For example, those provided by Lacombe [46],[45] and Grzegorczyk [30], who were also interested in the computable real-valued function around the same time as Kleene. Grzegorczyk and Lacombe wanted a definition that was as closely linked with classical analysis as possible. In 1955, they (independently) gave the following definition of a computable real-valued function; a function $f : \mathbb{R} \to \mathbb{R}$ should be considered computable if it is both *sequentially computable* (f maps every computable sequence of points into a computable sequence of points - recall Banach-Mazur computable functions!) and *effectively uniformly continuous* (there is a computable function $h : \mathbb{N} \to \mathbb{N}$ such that, for all x, y and all N, if we have $|x - y| < \frac{1}{h(N)}$ then $|f(x) - f(y)| < 2^{-n}$).

From the analytical standpoint, this is a natural definition, due to the fact that knowledge of a real-valued function on a dense set of points and continuity is sufficient to determine it. Grzegorczyk and Lacombe simply effectivise these two conditions [53].

We finish with one final, alternate notion by Caldwell and Pour-El,, who gave their classification in 1975. They defined a *computable sequence of*

polynomials to be a sequence defined by

$$p_n = \Sigma_{i=1}^{g(n)} r_{n,i} x^i,$$

where $g : \mathbb{N} \to \mathbb{N}$ is computable function and $(r_{n,i})_{n,i}$ a computable rational (double) sequence. They then call a function $f : \mathbb{R} \to \mathbb{R}$ computable if there exists a computable sequence of rational polynomials $(p_i(x))_i$ that converges effectively to f. For full details see [15]. This definition was proved equivalent to Grzegorczyk and Lacombe's in [53].

We could have used either of these definitions, or many of the others not stated here. However, in the effort to be as straight forward and consistent as possible, we have opted to for Kleene's Type II computable functions.

We now note a few interesting points about Type II computable functions. First, the reader will notice that Type II computable functions must be continuous over $\mathbb{R}$. Continuity is a result of the very nature of the Type II computable function, combined with the Use Principle. As a consequence, even simple functions like the following should not be considered computable!

$$f(x) = \begin{cases} 1 & \text{if } x \in \mathbb{Q} \\ 0 & \text{if } x \notin \mathbb{Q} \end{cases}$$

This seems reasonable, as how would we compute $f(x)$ if the rationality of x is not known (for example $x = \pi + e$)...

What about an even *simpler* discontinuous function, the sign function?

$$sgn(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$

For *sgn(x)* to be computable there would need to exist some algorithm to decide whether x is greater than, less than, or equal to 0. But we already know that, even if x is a *computable* real, we cannot decide whether or not $x = 0$. The problem becomes even harder if $x \in \mathbb{R} \backslash \mathbb{R}_c$. And so, it is not unreasonable that a computable real-valued function must be continuous.

This last example raises an interesting question. What about extending Borel computable functions to Type II computable functions? Is this always possible?

Obviously, based purely on domain differences, Markov/Borel and Type II computability are not the same. But what may not be as clear is that, even if we compare them with restricted or extended domains, these notions of computability remain (to some extent) distinct.

By a straightforward application of the Recursion Theorem, it is evident that any computable function restricted to $\mathbb{R}_c$ must be Borel computable. The converse is *not* true in general. We will give an example of a Borel computable function that cannot be extended to a Type II computable function. In fact, we give an example of a Borel computable function that cannot even be extended to any *continuous* function on $\mathbb{R}$.

Example 2.2.7. We will build a function $f : [0,1] \cap \mathbb{R}_c \to \mathbb{R}_c$ that is Borel computable, but cannot be extended to a continuous function on $\mathbb{R}$. Consider a standard enumeration of partial computable machines $\varphi_{e_1}, \varphi_{e_2}, \ldots$ that appear Cauchy (recall we built such a sequence in the proof of Theorem 2.2.3, Claim 2). Note that we are only interested in computable reals in the unit intervals, so discard all machines that approximate values outside of this. Call the mth term output by machine φ_{e_n} (if such a term exists) $q_{m,n}$. Notice that, partial or not, $q_{m,n}$ is a 2^{-m} rational approximation of some computable real. We now define f by a sort of diagonalisation process. We essentially set $f(x) = m$ for all $x \in (q_{m,m} - 2^{-m}, q_{m,m} + 2^{-m}) \cap \mathbb{R}_c$ not yet defined.

More formally, we construct f in stages. At stage n, let m be least such

that:

1. We have observed $q_{m,m}$

2. There exists an $x \in (q_{m,m} - 2^{-m}, q_{m,m} + 2^{-m}) \cap \mathbb{R}_c$ for which $f(x)$ has not yet been defined

Declare $f(x) = m$ for all x that satisfy 2.

Clearly f is Markov/Borel computable. The function f must be defined at every computable real by construction, so to decide $f(x)$, simply wait until the Cauchy name of x is entirely contained in an interval f is defined on.

There is a section of the unit interval that f is not defined on at every stage. This means, for all n there exists $x \in [0,1] \cap \mathbb{R}_c$ such that $f(x) = n$, i.e. on $[0,1]$ f attains arbitrarily large values. The Extreme value Theorem asserts that if a function is continuous on a bounded interval, it must attain a maximum and minimum on that interval [8].[13] Hence by the Extreme value Theorem, f cannot be extended to a continuous function on $\mathbb{R}$. Consequently, f cannot be extended to a Type II computable function. □

Note we could have given a similar example using Π_1^0 classes with no computable members.

Even though there exist Borel computable functions that are discontinuous almost everywhere on $\mathbb{R}$, if we restrict our attention to Type II computable functions we can (perhaps amazingly) compute their maximum and minimum on any interval.

The following result is an effectivisation of Bolzano's Extreme value Theorem. It can be found in [53]. We give our own proof of this result.

Theorem 2.2.8. *(Bolzano [8])(Extreme value Theorem) If a function*

---

[13]An English translation of this paper can be found in [9].

$f : \mathbb{R} \to \mathbb{R}$ *is continuous on a closed interval* $[a,b]$, *then* f *has both a maximum and minimum on* $[a,b]$.

Theorem 2.2.9. *(Pour-El and Richards [53])[14] Given a compact space* X *and a Type II computable function* $f : X \to \mathbb{R}$, $\max_{x \in X} f(x)$ *is computable.*

Proof: Let $X = [a,b]$, $a, b \in \mathbb{Q}$. We know $\max_{x \in [a,b]} f(x) := m$ exits by the original Extreme value Theorem, so to show that m is computable we need find its Cauchy name. It is sufficient to approximate m from above and below with two computable functions $g : \mathbb{N} \to \mathbb{Q}$ and $h : \mathbb{N} \to \mathbb{Q}$. We will define our functions by induction. Assume $g(x)$ and $h(x)$ are defined for all $x < n$. There exists a finite cover of of $[a,b]$ by Compactness. Also by Compactness, there exists a finite cover of Γ_f (the graph of f) of open balls of radius 2^{-n}. We want a special cover of $[a,b]$ such that the image of the cover of $[a,b]$ covers the graph Γ_f with open balls of radius at most 2^{-n}. This can be done computably by the Use Principle and is illustrated in Figure 2.1.[15]
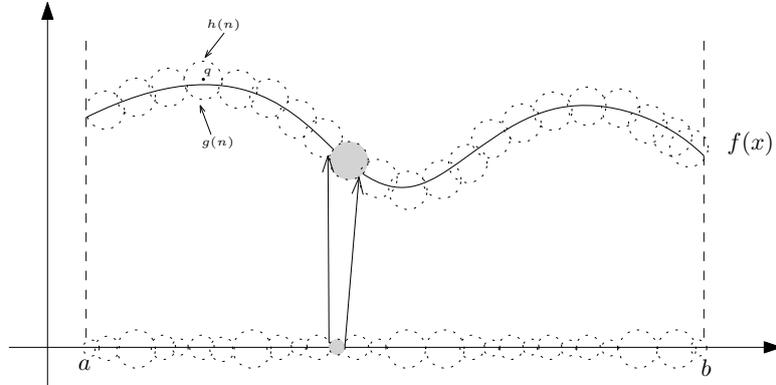
Figure 2.1: The open ball cover correspondence.

Once achieved, check the rational center point of every ball in the image

---

[14]It may be that this result appeared earlier that in the text given.

[15]The function f is Type II computable, so there exists an oracle machine Φ_e^x that outputs a Cauchy name for $f(x)$. If Φ_e^x reads the first m terms in a Cauchy name of x, and outputs a nth approximation of $f(x)$, then by the Use Principle, this is also a sufficient n^{th} approximation of $f(y)$ for any real $y \in (x_m - 2^{-m}, x_m + 2^{-m})$. That is, the interval $(x_m - 2^{-m}, x_m + 2^{-m})$ maps into the interval $(f(x)_n - 2^{-n}, f(x)_n + 2^{-n})$.

of the cover of $[a, b]$. Find the one with the greatest y-coordinate and let the y-coordinate of this point be q.

Define $g(n) = q - 2^{-n}$ if $g(n) \geq g(n-1)$, otherwise $g(n) = g(n-1)$.

Define $h(n) = q + 2^{-n}$ if $h(n) \leq h(n-1)$, otherwise $h(n) = h(n-1)$.

By the construction, h and g are both computable functions approaching m from above and below respectively, hence m has a computable Cauchy name. $\square$

We note that, although we can compute the maximum value f attains on a compact space, the point (or points) at which this maximum occurs may not be computable. Kreisel [39], Lacombe [47], and Specker [59] have all given examples of computable functions that do not reach a maximum at any computable real value.

Figure 2.2 summarises the relationship between the different types of computability we have covered in this chapter (see [3]).
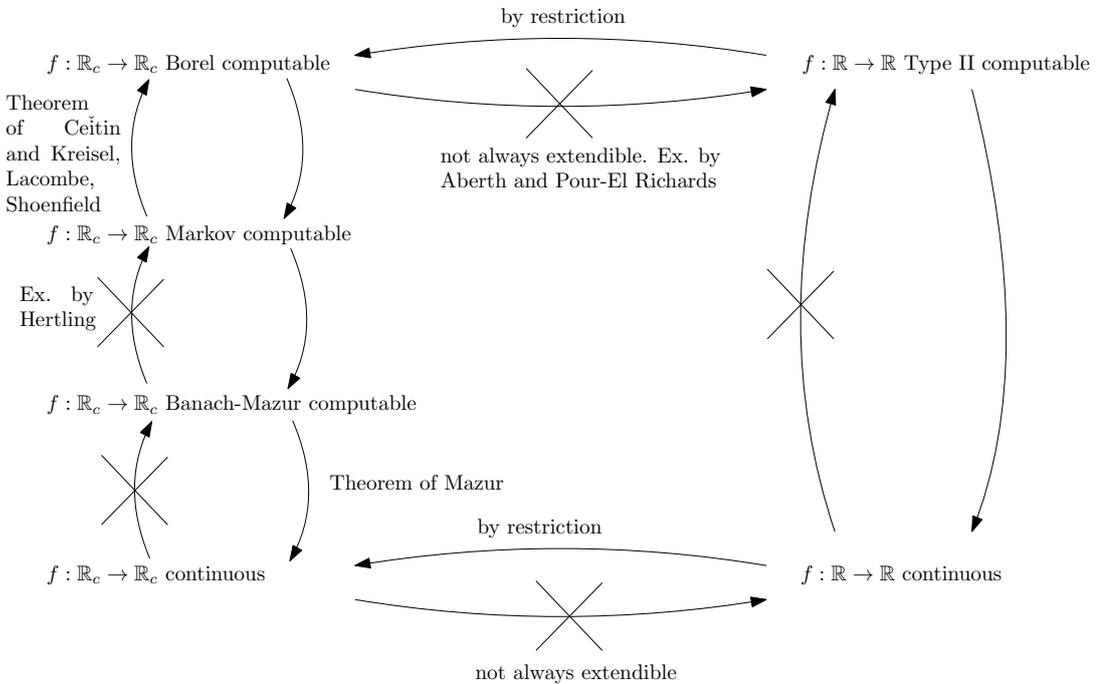

Figure 2.2: Function relationship summary.

Chapter 3

The Distance Function

So far we have considered computable reals and computable real-valued functions. We now ask what it means for a subset of $\mathbb{R}^n$ to be computable. We will define a new function, called the *distance function*, and call a compact set X computable if it has a computable distance function.[1] In this section, we focus on the computability of the distance functions of the graphs of Type II and Markov computable functions. We will then return to this notion in the final chapter when we discuss the Blaschke Selection Theorem. Note that Theorem 3.0.13 and Corollary 3.0.14 are new. The proof of Lemma 3.0.10 is original.

Recall that the *graph* of a function $f : D \subset \mathbb{R}^k \to \mathbb{R}^l$ is the set $\Gamma_f = \{(x, f(x))|x \in D\} \subset \mathbb{R}^{k+l}$.

Now that we are working with sets (rather than numbers or functions) we point out that it is sometimes useful to consider closed sets as Π_1^0 classes (and open sets as Σ_1^0 classes). Doing so allows us to utilise some of the tools of classical computability to prove our results. For example, we can prove that the graph of a Type II computable function $f : \mathbb{R} \to \mathbb{R}$ is a Π_1^0 class in Baire space.[2]

---

[1] As with the computable real and computable real-valued function, there exist other classifications of a computable subset of $\mathbb{R}^n$. See, for example, the Braverman and Yampolsky book [12].

[2] We emphasise that our result is for *Baire* rather than *Cantor* space. This does

Lemma 3.0.10. *If f is a Type II computable function, then Γ_f is a Π_1^0 class in Baire space.*

Proof: We will build a computable tree T is Baire space and define a string σ (which depends on (a,b)) such that $(a,b) \in \Gamma_f \iff (\forall n)$ $\sigma \upharpoonright n \in T$.

Let Φ^x be the oracle Turing machine that computes $f(x)$. We would like to build T so that the only paths in T are alternating rational Cauchy names for some x and $f(x)$. That is, if P is a path in T then there must exist a real a with Cauchy name $(a_i)_i$ such that $P(2n) = a_n$ and $P(2n+1) = f(a)_n$ (if $(f(a)_i)_i$ is the Cauchy name of $f(a)$ given by Φ^a). So we want $P = a_0 f(a)_0 a_1 f(a)_1 a_2 f(a)_2 \ldots$.

To ensure this, we begin by enumerating all possible rational sequences into T. At stage s we check all finite branches in T_s (T at stage s). Let $\tau_s \in T_s$ and $|\tau_s| = m$. Without loss of generality assume m is even. We kill this branch at stage s only if it satisfies one of two conditions:

1. The odd rationals in this string do not look Cauchy. That is, the sequence $\tau(1), \tau(3), \ldots, \tau(m-1)$ does not look like a Cauchy sequence.

2. There exists some even $n \leq \frac{m}{2}$ such that $\Phi^{(\tau(1),\tau(3),\ldots,\tau(m-1))}(n) = q$ and $|q - \tau(n)| > 2.2^{-\frac{n}{2}}$.

If 1. holds then the odd terms do not represent a Cauchy name of any real, so we kill the branch. If 2. holds then the even terms do not constitute a *potential* Cauchy name for some $f(x)$. Note we emphasise 'potential' because at some later stage the odd terms may fail to be Cauchy. Also notice that if $\Phi^{x_1, x_2, \ldots x_n}$ halts and outputs a rational q, this q must be a reasonable approximation of $f(x)$ by the Use Principle. Lastly, every

---

not have any effect in this thesis but is worth bearing in mind as our Π_1^0 classes are not computably bounded. As a result, classical theorems (for example the Low Basis Theorem) would not apply here.

pair of viable Cauchy names for the function f will be accepted as paths in the computable tree T.

Given a pair of Cauchy names $(a_i)_i$ and $(b_i)_i$ for $a, b \in \mathbb{R}$, define a string $\sigma \in \omega^\omega$ by letting $\sigma(2n) = a_n$ and $\sigma(2n+1) = b_n$. Then $(a, b) \in \Gamma_f \iff (\forall n)\ \sigma \upharpoonright n \in T$. Hence Γ_f is a Baire space Π_1^0 class as required. $\square$

We now define the distance function for a compact set C. Let

$$d_C(X) = \inf_{y \in C} |x - y| = \min_{y \in C} |x - y|,$$

where the second equality follows by compactness.

Definition 3.0.11. We say that a compact set C is *computable* if its distance function d_C is a Type II computable function.

For example, in $[0, 1]^2$ the set $C = [0, 1] \times \{0\}$ is a computable. Simply set $d_{[0,1]}((x, y)) = y$.

Sometimes the term *located* is used in the place of computable when describing these sets. Brouwer was the first to introduce this notion in a constructive setting [13]. He originally called these set "Katalogisiert", which means 'catalogued'.

We will now show that the graph of a Type II computable function on a bounded interval is located.

Theorem 3.0.12. *(Folklore) If $f : \mathbb{R} \to \mathbb{R}$ is a Type II computable function on a bounded interval then $d_{\Gamma_f} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a Type II computable function.*

Proof: To prove that d_{Γ_f} is computable, we show that the spaces above and below Γ_f are Σ_1^0 classes. Enumerating these connected components allows us to generate the sets of points strictly greater than, and less than, any fixed rational distance q from Γ_f. These sets can then be used to compute d_{Γ_f}

We first show that the connected components above and below Γ_f are Σ_1^0 classes. Note that we know these components are distinct because f is Type II computable, so continuous, and hence has a connected graph. Let (a, b) be any fixed point in our space. If $(a, b) \notin \Gamma_f$, then at some point open balls of decreasing radius, centred around some term in the pairs of Cauchy sequences $(a_i)_i$, $(b_i)_i$ and $(a_i)_i$, $(f(a)_i)_i$, become permanently separated. That is, there exists an n such that the open balls $B((a_n, b_n), 2^{-(n-1)})$ and $B((a_n, f(b)_n), 2^{-(n-1)})$ do not intersect.[3] Should we observe this, we know for sure that the corresponding point cannot be a member of Γ_f. Then we need only compare the two rational centres to determine whether the ball centred at (a_n, b_n) lies above or below Γ_f. More formally, the point (a, b) lies below $\Gamma_f \iff$ there exists an n such that $B((a_n, f(a)_n), 2^{-(n-1)}) \cap B((a_n, b_n), 2^{-(n-1)}) = \emptyset$ and $b_n < f(a)_n$. This is a Σ_1^0 condition. Similarly for any point above Γ_f.

Now we know that:

1. $d_{\Gamma_f}((x, y)) < q$, for $q \in \mathbb{Q}$, if and only if $B((x, y), q)$ intersects the class of points both above and below Γ_f.

2. $d_{\Gamma_f}((x, y)) > q$ if and only if $\overline{B}((x, y), q)$ is entirely contained in the complement of Γ_f.

Using these two facts, and the enumeration of the collection of points above and below Γ_f, for any $q \in \mathbb{Q}$ we can enumerate the collections of points $\{(x, y) : d_{\Gamma_f}((x, y)) < q\}$ and $\{(x, y) : d_{\Gamma_f}((x, y)) > q\}$; For every point (x, y), wait for an M that is far enough along in the respective Cauchy sequences $(x_i)_i$, $(y_i)_i$, such that $n, m > M$ implies both $|x_n - x_m| < \frac{q}{2}$ and $|y_n - y_m| < \frac{q}{2}$. We now enumerate the sequence of open balls $B((x_n, y_n), q)$ for all $n > M$, and the space above and below Γ_f. If there exists an n such that points in $B((x_n, y_n), q)$ appear in both spaces, then $d_{\Gamma_f}((x, y)) < q$. Similarly, if an open ball containing $\overline{B}((x_n, y_n), q)$, for some $n > M$, is enumerated into the space above or below Γ_f, we

---

[3]We can not just take the open balls $B((a_n, b_n), 2^{-(n)})$ and $B((a_n, f(a)_n), 2^{-(n)})$ here because the point (a, b) and $(a, f(a))$ may actually fall outside of these balls.

know that $d_{\Gamma_f}((x, y)) > q$. In this way we can enumerate the sets of points strictly greater than, and less than, any fixed rational distance q from Γ_f

Finally, we can now show that $d_{\Gamma_f}((x, y))$ is Type II computable. We provide a summary of the method, then follow with the precise details. Guess a distance q, and generate the two sets of points strictly greater than, and less than, q from Γ_f. Recall (a, b) is any fixed point in our space. We will use it here to demonstrate we can compute $d_{\Gamma_f}((a, b))$. Our given point (a, b) must occur in one of these sets eventually. If the distance between Γ_f and (a, b) is greater that q, repeat for $q + r$ for some appropriate rational r. If the distance is smaller than q, repeat instead for $q - r$. Choosing our distances sensibly, we will eventually bounce between two rationals q_1 and q_2. These rationals we can refine until $|q_1 - q_2| < 2^{-n}$, for any desired n. Then the computable function $g(n) = \frac{q_1 + q_2}{2}$ would sufficiently approximate $d_{\Gamma_f}((a, b))$ to within 2^{-n}. The precise details follow.

Suppose we want to approximate $d_{\Gamma_f}((a, b))$ to within 2^{-n}. We build a functional Φ with oracle $((x, y)_i)_{i \in \mathbb{N}} = (x_{i+1}, y_{i+1})_{i \in \mathbb{N}}$, a Cauchy name of (x, y), to compute this approximation. On input n, find rationals q_1, q_2, and terms in the given rapidly converging Cauchy sequences x_m, y_m, such that:

1. $|q_1 - q_2| < 2^{-(n+1)}$

2. $d_{\Gamma_f}((x_m, y_m)) < q_2$

3. $d_{\Gamma_f}((x_m, y_m)) > q_1$

4. $|(x, y) - (x_m, y_m)| < 2^{-(n+2)}$

So, we have $q_1 - 2^{-(n+2)} < d_{\Gamma_f}((x, y)) < q_2 + 2^{-(n+2)}$.

Let $\Phi^{(x,y)}(n) := \frac{q_1 + q_2}{2}$. Then,

$$
\begin{aligned}
|d_{\Gamma_f}((x,y)) - \Phi^{(x,y)}(n)| &= |d_{\Gamma_f}((x,y)) - \frac{q_1 + q_2}{2}| \\
&< |q_2 + 2^{-(n+2)} - \frac{q_1 + q_2}{2}| \\
&= |2q_2 + 2^{-(n+1)} - (q_1 + q_2)| \\
&= |q_2 - q_1 + 2^{-(n+1)}| \\
&\leq |q_2 - q_1| + |2^{-(n+1)}| \\
&< 2^{-(n+1)} + 2^{-(n+1)} \\
&= 2^{-n}
\end{aligned}
$$

Setting $(x,y) = (a,b)$ allows us to compute $d_{\Gamma_f}((a,b))$. Therefore, $d_{\Gamma_f}((x,y))$ is Type II computable. $\square$

Notice that the effective Extreme value Theorem (Theorem 2.2.9 from the previous section) is now an easy Corollary of this result.

We now ask what happens if we instead consider a Markov computable function defined on the computable reals. Is the distance function of Γ_f also computable? It turns out that this is not the case.

Theorem 3.0.13. *A Markov computable function $f : \mathbb{R}_c \to \mathbb{R}_c$ on a bounded interval has an upper semi-computable, but not necessarily Type II computable, distance function d_{Γ_f}.*

The proof will follow in two parts. Part one will show that $d_{\Gamma_f}((x,y))$ is upper semi-computable, and part two that $d_{\Gamma_f}((x,y))$ is not computable. Recall by Theorem 2.2.3 we know that Borel and Markov computability are equivalent so we can use these two notions interchangeably.

Proof Part 1: First recall that a partial function $f : \mathbb{R}_c \to \mathbb{R}$ is *upper semi-computable* (which means it can be approximated from above) if there exists a computable function of two variables $\phi(x,k) : \mathbb{R}_c \times \mathbb{N} \to \mathbb{R}_c$ where x is the desired parameter for $f(x)$ and k the level of approximation

such that:

1. $\lim_{k \to \infty} \phi(x, k) = f(x)$

2. $\forall k \in \mathbb{N} : \phi(x, k+1) \leq \phi(x, k)$

Fix $(a, b) \in \mathbb{R}_c^2$. Recall that we consider the distance function $d_{\Gamma_f}(x, y)$ computable if we can computably give a computable Cauchy name for every pair of points (x, y). We are not trying to show that d_{Γ_f} is computable, but rather upper semi-computable. So instead of computably giving a Cauchy name for every input, we want to build a computable function $g(n, k)$ that approximates (from above) every term in the Cauchy name of *fixed* (a, b). Then, if we call $d_{\Gamma_f}(n, (a, b))$ an approximation to the true distance between (a, b) and Γ_f with an accuracy of 2^{-n}, $\lim_{k \to \infty} g(n, k) = d_{\Gamma_f}(n, (a, b))$. Taking both k and n to infinity then achieves the desired result; $\lim_{n,k \to \infty} g(n, k) = d_{\Gamma_f}((a, b))$. Note we emphasise that a different $g(n, k)$ must be constructed for *every* pair of points $(a, b) \in \mathbb{R}_c^2$.

We will show by induction how to define $g(n, k)$ for fixed $(a, b) \in \mathbb{R}_c^2$. Let the function $g(m, k)$ be defined for all $m < n$. Call $d_{\Gamma_f, n}((a, b))$ the upper bound of the nth approximation of $d_{\Gamma_f}((a, b))$ (note that this exists by the inductive hypothesis). We will define a sequence $(g(n, k))_k$ that approaches $d_{\Gamma_f, n}((a, b))$ from above. This is done by finding the distance between an appropriately close approximation of $(x, f(x))$ and (a, b) for every computable real x (which depends on n), and defining $g(n, s)$ to be the least of these distances at each stage. This will ensure $(g(n, k))_k$ approaches $d_{n, \Gamma_f}((a, b))$ from above, and ultimately $\lim_{n,k \to \infty} g(n, k) = d_{\Gamma_f}((a, b))$. The precise details of the construction follow;

Initially we wait for terms $f(a)_{n+2}$ and b_{n+2} in the respective Cauchy names of $f(a)$ and b such that $|f(a)_{n+2} - f(a)| < 2^{-(n+2)}$ and $|b_{n+2} - b| < 2^{-(n+2)}$. Set $g(n, 0) = |f(a)_{n+2} - b_{n+2}| + 2^{-(n+1)}$.[4]

---

[4]Note that $d_{\Gamma_f, n}((a, b)) \leq |f(x)_{n+2} - y_{n+2}| + 2^{-(n+1)} = g(n, 0)$.

Let $e_1, e_2 \ldots$ be a listing of all partial machine indices. Assume we are at stage s, n is fixed, and Φ^x is the oracle Turing machine that approximates the Markov computable function f. We assume (again, by induction) that $g(n, t)$ has been defined, and will now define $g(n, t + 1)$. Call an index 'active' if it was not 'killed' at an earlier stage. For least active e_i, we ask whether or not φ_{e_i} has output a finite Cauchy name $q_1, \ldots, q_m$ after being run for s stages. If this is not the case, discard this index for all future stages, and check the next active index. If on the other hand $q_1, \ldots, q_m$ does appear to be Cauchy, first note that $q_1, \ldots, q_m$ looks like the initial terms of the Cauchy names of a *range* of computable reals (specifically, any $q \in (q_m - 2^{-m}, q_m - 2^{-m})$). We assume, without loss of generality, $m > n + 3$ (if not we can wait until a later stage where this is the case and the sequence looks Cauchy). We then ask whether this sequence, used as an oracle in Φ, is sufficient to compute what looks like a Cauchy approximation of $f(q)$, to within $2^{-(n+3)}$. This means we run $\Phi^{q_1, \ldots, q_m}$ for s stages and, if $\Phi^{q_1, \ldots, q_m}$ outputs a sequence $r_1, \ldots, r_{n+3}$ that looks Cauchy, we have a success!

If we do not have success, repeat steps above for stage $s + 1$. If we do have a success, we now have an approximation (q_m, r_{n+3}) which is within $2^{-(n+3)}$ of some computable real $(q, f(q))$ (in fact a range of such pairs). Note that we are confident of this because we assumed $m > n + 3$.

Next, calculate the distance D between (a_{n+3}, b_{n+3}) and (q_m, r_{n+3}). This is summarised in Figure 3.1. Recall that, by the inductive assumption, we have defined g at this point up to $g(n, t)$. If $D + 2^{-(n+1)} \leq g(n, t)$, set $g(n, t+1) = D + 2^{-(n+1)}$. If not, set $g(n, t+1) = g(n, t)$. Notice that we need to add $2^{-(n+1)}$ to D to ensure that we approach the 2^{-n} distance approximation from *above*. We now 'kill' the index e_i for this *particular fixed* n and go to stage $s + 1$. We emphasise that stage $s + 1$ is still operating with the *same* fixed n, and will defined $g(n, t + 2)$. When we change n (which involves a completely separate construction) we must then reset all indices.

This construction will give us a sequence $(g(n, k))_k$ that approaches
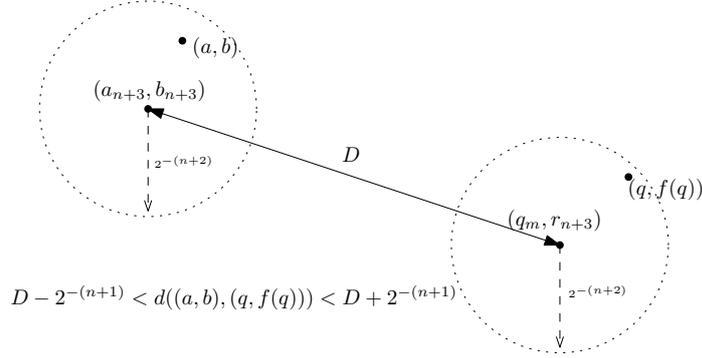
Figure 3.1: The distance D.

$d_{n,\Gamma_f}((a,b))$ from above. We also run this construction for other n, building next, for example, the sequence $(g(n+1,k))_k$. As mentioned, all indices must be reset every time n is updated.

This construction gives us a computable sequence $\bigcup_{n,k}(g(n,k))_k$ such that for all n and k:

$g(n,k) \geq d_{n,\Gamma_f}((a,b)),$

$g(n,k) \geq g(n,k+1)$

and

$\lim_{n,k\to\infty} g(n,k) = d_{\Gamma_f}((a,b)).$

Therefore $d_{\Gamma_f}((a,b))$ is upper semicomputable. $\square$

Proof Part 2: For the second part of the theorem we will show that, for any noncomputable right c.e. real α, there exists a Markov computable function f such that $d_{\Gamma_f}((0,0)) = \alpha$. The origin is chosen for simplicity, but the proof works just as well for any $(p,q) \in \mathbb{R}_c^2$.

Recall that a *right c.e. real* is a real $x \in \mathbb{R}$ and c.e. sequence $(x_i)_i$ such that $\lim_i x_i = x$ and $(\forall i)(x_i \leq x_{i-1}$ and $x_i > x)$.

Let $q_1, q_2, \ldots$ be noncomputable c.e. sequence converging to α from above and assume, without loss of generality, $\alpha < 1$. We will define the Markov

computable function f in stages. At each stage n we define f for at least all values $x > v$ (which v chosen at each stage is specified in the construction). Informally, we would like to define $f(q_i) = 0$ for all i, and almost everywhere else have $f(x) > 0$. In particular, we will allow $f(x) = 0$ only if $x \geq q_i$ some q_i in $(q_i)_i$. This will help to ensure that $d_{\Gamma_f}((0,0)) = \alpha$. The function f also needs to be Markov computable, which means for any $u \in \mathbb{R}_c$ we need to be able to evaluate $f(u)$ (given an approximation of u we can find an approximation of $f(u)$). By Claim 2 in Theorem 2.2.3 (existence of c.e. set W) this means that, given an open ball $B \ni u$ of any radius, we need to map B under f into another open ball B' such that: B' contains $f(u)$, and for all $x \in B \cap \mathbb{R}_c$ we have $f(x) \in B'$.

We ensure this by doing the following; if an open ball B containing u falls into a range of values already defined at the current stage, we simply evaluate B at f and refine its radius to achieve the desired approximation of $f(u)$. If B falls outside the defined values, we essentially set $f(x) = 1$ for all $x \in B$, and incorporate this into our construction at some later stage. Whenever we define f on an interval $[a, b]$, we always ensure that $f(a) = f(b) = 1$ (if this is instead an open interval then we simply have $f(x)$ tending to 1 as x tends to a from the right, and b from the left). This is done in a consistent manner to ensure f is continuous.

The formal construction follows.

Stage 0: Observe the first term q_0 in the rational noncomputable c.e. sequence converging to α from above. For simplicity assume $q_0 \leq 1$.

Stage 1: Wait for next term q_1 in the sequence. q_0 and q_1 are rationals, so let $|q_0 - q_1| = b_1 \in \mathbb{Q}^+$ and $b_0 = 1$. Define f to vary linearly from 1 to 0 from $[q_1 + \frac{b_1}{2}, q_0]$, and 0 to 1 from $[q_0, q_0 + b_0]$. Call $v_1 := q_1 + \frac{b_1}{2}$.[5]

Stage n: Wait for the next term q_n in the sequence. Let v_{n-1} be the least rational f has been defined at such that, for all $x > v_{n-1}$, the value $f(x)$ is defined at this stage. Check $d(q_{n-1}, v_{n-1})$, and $d(q_i, q_{i-1})$

---

[5]If we were asked to evaluate u at this stage, defer to Stage 2.

for all $i \leq n$, and let the least of these distances be D. Choose a k such that $2^{-k} < \frac{D}{4}$. We do this to ensure that if we need to evaluate $f(u)$ for $u \in \mathbb{R}_c$, $B(u_k, 2^{-k})$ contains at most one element from the observed sequence $q_1, \ldots, q_n$ tending to α

Step 1: If we do not need to evaluate $f(u)$ at this stage, go to Step 2. Otherwise, we want to evaluate $f(u)$ at this stage, $u \in \mathbb{R}_c$, with Cauchy name $(u_i)_i$. There three sub-cases to consider. They occur as combinations of two conditions.

Condition (a) For all $x \in B(u_k, 2^{-k})$, $f(x)$ has not yet been defined.

Condition (b) There exist $r_1, r_1 \in \mathbb{Q}^+$ such that for all x, $r_1 < x < u_k - 2^{-k}$ and $u_k + 2^{-k} < x < r_2$, $f(x)$ has not yet been defined.

    I If both (a) and (b), set $f(x) = 1$ for all $x \in B(u_k, 2^{-k})$ and go to Step 2.

    II If (a) but not (b), wait for $k' > k$ such that (b) holds, then set $f(x) = 1$ for all $x \in B(u'_k, 2^{-k'})$ and go to Step 2.

    III If not (a), then $f(x)$ defined on some points in $B(u_k, 2^{-k})$ already. Wait for $k' > k$ such that (a) applies to $B(u_{k'}, 2^{-k'})$ OR $\forall x \in B(u'_k, 2^{-k'})$ $f(x)$ has already been defined. In the first case set $f(x) = 1$ for all $x \in B(u_{k'}, 2^{-k'})$ and go to Step 2. In the second, do nothing, go to Step 2.

We allow at most one such calculation at each stage.

Step 2: Observe q_n. Either at some earlier stage we were asked to evaluate $w \in \mathbb{R}_c$, and consequently defined for some m all $x \in B(w_m, 2^{-m})$ including $q_n \in B(w_m, 2^{-m})$, or not. (We can decide this computably as we have been asked to evaluate only finitely many computable reals at this stage).

    IF NO: Recall that v_{n-1} is the least rational f was defined at such that $\forall x > v_{n-1}$, $f(x)$ has been defined.

I If there exists some x such that $q_n < x < v_{n-1}$, and $f(x)$
   has already been defined, let c be greatest such that if
   $q_n < x < c$, $f(x)$ has not yet been defined. Let $b =$
   $q_n + 3\frac{c-q_n}{4}$ and $a = q_n + \frac{c-q_n}{2}$. Then set $f(x) = 1$ for all
   $x > c$ not yet defined. Let $f(x)$ vary linearly from 1 to 0
   on $[a,b]$ and from 0 to 1 on $[b,c]$. $f(x)$ is now defined for
   at least all $x > a$. This process is summarised in Figure
   3.2.
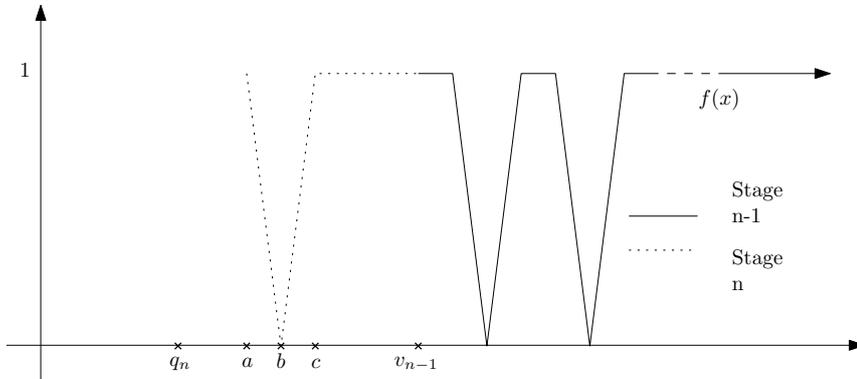
Figure 3.2: The construction of f if 'No : I'.

II Otherwise, do as in I, but let $c = q_n + 3\frac{v_{n-1}-q_n}{4}$, $b =$
   $q_n + \frac{v_{n-1}-q_n}{2}$ and $a = q_n + \frac{v_{n-1}-q_n}{4}$.

IF YES: First, for all $x > w_m - 2^{-m}$ not yet defined, let $f(x) = 1$ (note
   that this preserves continuity). Find smallest rational d such
   that $d > 0$ and for all x, $d < x < w_m - 2^{-m}$, $f(x)$ has not
   yet been defined. Such a d must exist by Step 1, Condition
   (b). Choose some rational γ such that $d < \gamma < w_m - 2^{-m}$.
   Let $f(\gamma) = \sqrt{q_n^2 - \gamma^2}$ and $f(x)$ to vary linearly from 1 to $f(\gamma)$
   on $[d + \frac{\gamma-d}{2}, \gamma]$, and from $f(\gamma)$ to 1 on $[\gamma, w_m - 2^{-m}]$. $f(x)$
   is now defined for at least all $x > d + \frac{\gamma-d}{2}$. This process is
   summarised in Figure 3.3.
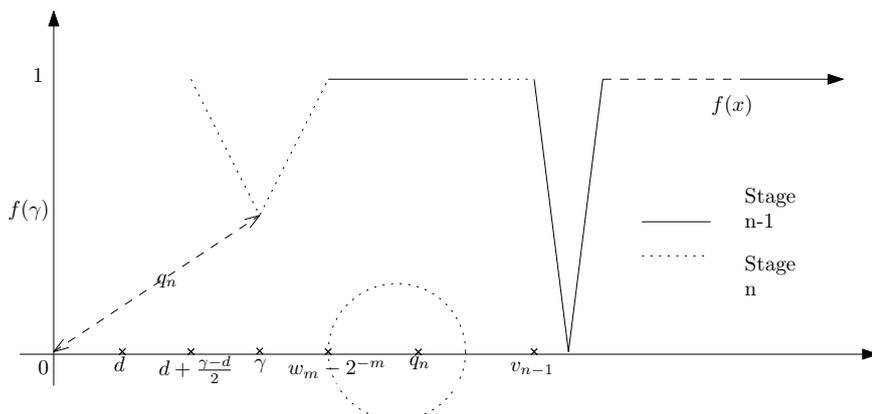
Figure 3.3: The construction of f is 'Yes'.

Justification:

We first note that f is continuous. Whenever we defined f on an interval $[a, b]$, we ensured $f(a) = f(b) = 1$. Each consecutive stage then essentially involved connecting some these intervals, and extending f to be defined at every point above some rational v_s. Continuity was preserved for all values above v_s, for each s, and every time we defined f below the current v_s, we ensured sufficient undefined space to ensure continuity at all later stages.

Claim: f is Markov computable.

Proof of claim: Suppose at stage n we wish to evaluate $f(u)$ for some $u \in \mathbb{R}_c$. Find $B(u_k, 2^{-k})$ as specified in construction. Then we either define $f(x) = 1$ for all $x \in B(u_k, 2^{-k})$, in which case $f(u) = 1$, or else there exists a k' such that f at all $x \in B(u'_k, 2^{-k'})$ has already been defined. In this case we can refine k' until we achieve the desired approximation of $f(u)$.

So f is a Markov computable function, and

$$\lim_{n \to \infty} d_{\Gamma_f}((0, 0)) = \lim_{n \to \infty} (q_i)_{i \in \mathbb{N}} = \alpha.$$

Hence, d_{Γ_f} is not computable. $\square$

Notice that $(0,0)$ is a computable point, and we have shown $d_{\Gamma_f}((0,0)) \notin \mathbb{R}_c$. This means d_{Γ_f} cannot be Markov computable either.

Corollary 3.0.14. *A Markov computable function $f : \mathbb{R}_c \to \mathbb{R}_c$ on a bounded interval does not necessarily have a Markov computable distance function d_{Γ_f}.*

Chapter 4

The Darboux Property

4.1   Introduction

The Intermediate value Theorem states that a continuous function $f :
[a, b] \to \mathbb{R}$ takes every value between $f(a)$ and $f(b)$ [7].[1] By isolating this
property, we define a new class of functions that are called Darboux.

Definition 4.1.1. A function f defined on an interval I has the *Darboux
property* if for all $a < b$, $a, b \in I$ and all $y \in [f(a), f(b)]$ there exists a
$x \in [a, b]$ such that $f(x) = y$.

Sometimes the Darboux property is instead called the *intermediate value
property*. We call the class of functions with the Darboux property $\mathcal{D}$,
and the class of continuous functions $\mathcal{C}$. This chapter will be dedicated
to exploring some of the characteristics of $\mathcal{D}$.

Before the late 19th century, the intermediate value property (as it was
known at the time) was given as part of the definition of a continuous
function. In fact, many mathematicians assumed that this property and
continuity were equivalent. It was not until 1875 that the French math-
ematician Jean Gaston Darboux gave a proof that every derivative has

---

[1]An English translation of this paper can be found in [9].

the Darboux property, and as not every derivative is continuous, separated the two classes $\mathcal{D}$ and $\mathcal{C}$ [19]. Why does this interest us? The Darboux property defines a strange class of functions, so this fact alone makes its complexity worth investigating. However, it turns out that *every* real-valued function is the limit of a sequence of Darboux functions (Sierpiński [56]). We are interested in how complicated these sequences are, and their relationship with computable real-valued functions.

Before we begin discussing Sierpiński's result, we first take a moment to further discuss the Darboux property. We give a proof that every derivative has the Darboux property. The following is a common version of this proof, and can be found in [62].

Theorem 4.1.2. *(Darboux [19]) Let I be an interval. If $f : I \rightarrow \mathbb{R}$ is differentiable on I then f' (the first derivative of f) has the Darboux property.*

Proof: Let f be a differentiable function. We will show that f' has the Darboux property. Let $a, b \in I$, where $a < b$, and $r \in \mathbb{R}$, where r is between $f'(a)$ and $f'(b)$. Without loss of generality, assume $f'(a) < r < f'(b)$. We now define a function $g : [a, b] \rightarrow \mathbb{R}$. Let

$$g(x) = f(x) - rx \qquad \text{for } x \in [a, b].$$

By a well known result in calculus, if $f(x)$ is differentiable then it is continuous. rx is trivially continuous, and as the difference of two continuous functions is continuous (another classic result, see any standard calculus textbook), $g(x)$ must also be continuous. By the Extreme value Theorem, there must exist a point $c \in [a, b]$ such that $g(c)$ is a maximum on $[a, b]$. By properties of derivatives, this means $g'(c) = 0$. Differentiating both sides of $g(x) = f(x) - rx$ and substituting c for x gives

$$g'(c) = f'(c) - r$$
$$f'(c) = r$$

The result follows. □

Consequently, continuity is not a necessary condition for membership of $\mathcal{D}$. A basic example of a function $f : \mathbb{R} \to \mathbb{R}$ with a discontinuous derivative is

$$f(x) = \begin{cases} x^2 \sin(\frac{1}{x}) & \text{if } x \neq 0 \\ 0 & x = 0 \end{cases},$$

with the derivative

$$f'(x) = \begin{cases} 2x \sin(\frac{1}{x}) - \cos(\frac{1}{x}) & \text{if } x \neq 0 \\ 0 & x = 0 \end{cases}.$$

f' must have the Darboux property, but is discontinuous at a single point; $x = 0$.

Of course, f's derivative is not particularly discontinuous, but there do exist functions with extremely discontinuous derivatives. As a result, even those functions that are discontinuous on every interval can belong to $\mathcal{D}$. We will give examples of these kinds of functions in the sections that follow. It is the fact that these types of functions are Darboux that allows us to easily approximate every real-valued function. Investigating how complicated these sequences are is the focus of a large part of this chapter, but we also look at how the complexity of these Darboux sequences changes as we restrict the domain and range of the principal function (the function we are interested in approximating).

In section 4.2 we discuss how to approximate a real-valued function with a sequence of Darboux functions. We then comment on the complexity and computational power of particular types of Darboux functions that take every real value on every interval. In section 4.3 we explore the Darboux property when restricted to $\mathbb{Q}$, and the effect this has on the complexity of approximating sequences. Lastly, in section 4.4 we discuss

uniformly convergent sequences of Darboux functions (in contrast to the usual pointwise convergence applied in earlier sections) and give examples of real-valued functions that cannot be uniformly approximated by Darboux functions.

4.2   Approximating real-valued functions

In 1953, Sierpiński proved that every real-valued function $f : \mathbb{R} \to \mathbb{R}$ is the limit of a sequence of Darboux functions [56]. We will first sketch a proof of this result, and then investigate how hard it is to construct such a sequence for any given function. We would also like to apply our result to Type II computable functions. However, notice that every Type II function is continuous, and so is the limit of a trivial sequence of Darboux functions (namely, itself). So instead, we will mention how hard is it to construct a non-trivial Darboux sequence for any Type II computable function. Theorems 4.2.10, 4.2.11, 4.2.17, 4.2.18, Corollaries 4.2.12 and Lemmas 4.2.14, 4.2.15 are original.

4.2.1   The closure of the class $\mathcal{D}$ is all functions

In order to sketch a proof of Sierpiński's result, it is first helpful to construct a function that maps every interval to all of $\mathbb{R}$. This is a (strong) example of a Darboux function that is discontinuous on every interval, and a similar function will be used in the proof to follow. We will call these types of functions canonical Darboux functions.

Definition 4.2.1. A function $f : I \to I$ (where I is any interval, including the entire real line) is a *canonical Darboux function* if for every interval $J \subseteq I$, $f(J) = I$.

There are a number of examples of this kind, and many of them can be adapted to prove Sierpiński's result. For example, Conway constructed an extreme base 13 function which takes every real value on

every nonempty open interval.[2] Radcliffe also gave a similar result in [54] using the function $f(x) = tan(n\pi x)$. The following example can be found in [62].

We construct a canonical Darboux function $f : [0, 1] \to [0, 1]$.

Example 4.2.2. Let D be the set of all rational numbers in the unit interval with finite decimal expansion. That is, $D := \{m10^{-n} : m \in \mathbb{Z}, n \in \mathbb{N}, n \geq m\}$. If $t \in [0, 1]$ has the expansion $t = 0.t_1t_2t_3\ldots$, $t_i \in \mathbb{N}$ where $t_k = 0$ for all sufficiently large k, if $t \in D$ then define $t^* = 0.t_1t_1t_2t_1t_2t_3\ldots$. This means t^* takes t and outputs the first decimal point in t, followed by the first and second, and so on. We can think of each of these repeating 'blocks' as initial segments of t, increasing by one bit every time a new block is added.

Notice that if $t, s \in [0, 1]$ are distinct, it follows that $t^* - s^* \notin D$ because t^* and s^* will differ at infinitely many places. In addition, for every $x \in [0, 1]$ there exists at most one $t \in [0, 1]$ such that $x - t^* \in D$.

We can now define our function.

$$f(x) = \begin{cases} t & \text{if } t \in [0, 1] \text{ and } x - t^* \in D \\ 0 & \text{otherwise} \end{cases}$$

To verify that f is a canonical Darboux function, let $I = [a, b]$ be any subinterval of the unit interval. For any $t \in [0, 1]$, there exists an $x \in I$ such that $x - t^* \in D$ (enumerate something in the given interval until a stage where any real extending this enumeration must be contained in the interval, and from this stage onwards copy t^*). Hence, for all $t \in [0, 1]$ there exists an $x \in I$ such that $f(x) = t$. It follows immediately that f must have the Darboux property.

The following proof can be found in [62]. We repeat it here and later analyse the complexity of this result.

---

[2]He formed this example while preparing for lectures [18].

Theorem 4.2.3. *(Sierpiński [56]) Every real-valued function $f : \mathbb{R} \to \mathbb{R}$ is the limit of a sequence of functions with the Darboux property.*

Proof sketch: The proof of this theorem follows by constructing a function very similar to the previous one. Let D and t^* be as above. Define a set $V := \{m5^{-n} : m \in \mathbb{Z}, n \in \mathbb{N}\}$, and collection of sets $V_i := V + 2^{-i}$. $V_1, V_2, \ldots$ are pairwise disjoint subsets of D (because 2^{-i} is an unique infinite repeating sequence when expressed in base 5). $\forall x \in [0, 1]$ there exists at most one $t \in [0, 1]$ such that $x - t^* \in V_i$.

We then define a sequence of functions $f_1, f_2, \cdots : [0, 1] \to [0, 1]$.

$$f_i(x) = \begin{cases} t & \text{if } t \in [0, 1] \text{ and } x - t^* \in V_i \\ f(x) & \text{otherwise} \end{cases}$$

Every f_i has the Darboux property, and $\lim_{i \to \infty} f_i = f$. $\square$

In the next subsection, we discuss the complexity of these kinds of sequences. As far as we know, every proof in the literature of the Sierpiński's result involved the construction of not just a sequence whose terms had the Darboux property, but in fact, the *canonical* Darboux property. This property is stronger than strictly necessary, and these functions are, as a consequence, quite complex. It remains open whether *every* real-valued function is the limit of a sequence of *non*-canonical Darboux functions.

4.2.2   The complexity of canonical Darboux functions

In this subsection, we discuss the complexity of the canonical Darboux function. We want to decide how much computational power is sufficient (or insufficient) to compute certain Darboux functions. To avoid confusion, we will first explain exactly what is intended by 'sufficient

computational power'. Recall that $x' = \{e : \phi_e^x(e) \downarrow\}$. We define a function that will in some sense correspond to the 'jump' relative to $\mathbb{R}$.

Definition 4.2.4. Let $J : \mathbb{R} \to \mathbb{R}$ denote a function that uniformly maps $x \mapsto x'$. Call this function the *jump operator* and refer to it as J.

We can generalise this definition and call the function $J^{(2)} : \mathbb{R} \to \mathbb{R}$ that uniformly maps $x \mapsto x''$ the *double jump operator* and refer to it as $J^{(2)}$ (and so on for n^{th} jump operator $J^{(n)}$).

In the work to follow, when dealing with the complexity of Type II functions, we consider the jump operator as a Type II functional. We will measure the computational strength sufficient to compute particular functions in terms of $J^{(n)}$. Later, when we address Markov computable and classically computable functions, it is enough to consider the 'jump' in the usual sense; as a set.

In this subsection, we will prove that J is insufficient to compute any canonical Darboux function. It then follows that J is insufficient to compute the terms in the sequences of canonical Darboux functions we constructed earlier to approximate any real-valued function. We will first classify a particular class of functions and give some necessary definitions.

A function $f : \mathbb{R} \to \mathbb{R}$ is *Baire class 0* if it is a continuous function. A function f is *Baire class 1* if f is the (pointwise) limit of a sequence of continuous functions $(f_i)_i$. In general, a function f is *Baire class n* if f is the (pointwise) limit of a sequence of Baire class $n-1$ functions $(f_i)_i$. We now adapt these definitions to a computable setting.

Definition 4.2.5. A function $f : \mathbb{R} \to \mathbb{R}$ is *effective Baire class 1* if $f(x) \leq_T J(x)$ uniformly in x.

More generally,

Definition 4.2.6. A function $f : \mathbb{R} \to \mathbb{R}$ is *effective Baire class n* if $f(x) \leq_T J^{(n)}(x)$ uniformly in x.

The detail given here regarding the Baire classes is sufficient for the purpose of this thesis, but for more information see [28] and [44].

Lastly, we define what it means for a real number x to be 1-generic. Recall the definition of a 1-generic element $x \in 2^\omega$. Let $W \subseteq 2^{<\omega}$. We say that x *meets* W if there is an initial segment σ of x such that $\sigma \in W$. x *avoids* a set W if there exists an n such that, for all τ extending $x \restriction n$, $\tau \notin W$.

Definition 4.2.7. An element $x \in 2^\omega$ is *1-generic* if it meets or avoids every c.e. set W.

Theorem 4.2.8. *An irrational $x \in [0,1]$ is 1-generic if and only if its unique binary expansion is 1-generic in 2^ω.*

A proof of this result can be found in [44]. Assume a real $x \in [0,1]$ is expressed as its unique binary expansion from this point onward. For rationals, refers to their dyadic expansion (if such a representation exists), although notice that rationals cannot be 1-generic, so uniqueness is not a problem.

Now that we have introduced the relevant concepts, we want to prove that J is insufficient to compute any canonical Darboux function. Or in other words, no canonical Darboux function is effective Baire class 1. This result is a corollary of the following theorem.

Theorem 4.2.9. *(Folklore) If x is a 1-generic real, and f is effective Baire class 1, then f is continuous at x.*

Proof: Suppose that $f(x) \leq_T J$. That is, $\exists e \; \forall x \; \Phi_e^{x'} = f(x)$. Let x be a 1-generic real. Notice that there is a continuous mapping between x and x' (by which we mean the set $\{e : \varphi_e^x(e) \downarrow\}$). This is because initial (although not necessarily computable) segments of 1-generic x determine initial segments of x', a unique property of 1-generics. There is also a continuous mapping between x' and $f(x)$ by assumption (simply map the *Use* $(f(x) \restriction n) \longmapsto f(x) \restriction n$), hence a continuous mapping from x to $f(x)$. $\square$

Our result is now a consequence of this theorem.

Theorem 4.2.10. *Let $f : [0,1] \to \mathbb{R}$ be a canonical Darboux function. Then f is not effective Baire class 1.*

Proof: This result follows directly from the theorem above. Specifically, if f is a canonical Darboux function then it is discontinuous everywhere, and therefore cannot be continuous at 1-generic reals. $\square$

As a result, the jump function is not strong enough to carry out the construction of Theorem 4.2.3. To do so, we need more computational power. However, it turns out that $J^{(2)}$ is sufficient to compute at least *some* canonical Darboux functions. Example 4.2.2 is such a function.

Theorem 4.2.11. *There exists a canonical Darboux function $f : [0,1] \to \mathbb{R}$ that is also effective Baire class 2.*

Proof: We reconstruct the function from Example 4.2.2 to illustrate this.

Usually, the argument would begin by assuming we have a Cauchy name for some real $x \in [0,1]$, however, the construction of this function is easier if we instead have a binary approximation of x. These notions are not usually equivalent (as noted by Turing in his correction [61]), however the double jump function can build a binary sequence that sufficiently approximates a Cauchy name of x. We will do this first.

Let $x_1, x_2, \ldots$ be a Cauchy name for $x \in [0,1]$. When we build our binary approximation of x, we also need to ensure we avoid creating duplicates. That is, we must make sure that f maps any two Cauchy sequences that are converging to x, to the same value. We will express x finitely in binary, if possible, to ensure this.

Given Cauchy name $x_1, x_2, \ldots$, the double jump function can decide whether x is a dyadic rational. If x is *not* a dyadic rational, then for all n and $m \leq n$, we can decide in finite time whether $x \in (m2^{-n}, (m+1)2^{-n})$. We build a binary expansion $0.b_1 b_2 b_3 \ldots$ of x as follows. We begin by asking if $x \in (0, \frac{1}{2})$. If $x \neq \frac{1}{2}$ then at some stage we will see the Cauchy

name of x fall entirely inside, or outside, of this interval. If the former, we set b_1, the first binary point of x, to be 0. If the latter, set $b_1 = 1$. We then repeat this process; if, (for example), $b_1 = 0$ and $x \neq \frac{1}{4}$, we ask whether $x \in (0, \frac{1}{4})$. If yes, set $b_2 = 0$, if no, then $x \in (\frac{1}{4}, \frac{1}{2})$ set $b_2 = 1$, and so on. In this way, we obtain an accurate binary expansion of $x = 0.b_1 b_2 b_3 \dots$. A problem of course could occur if $x = \frac{1}{2}$, or indeed $x = \frac{a}{2^b}$ for any natural a and b. It follows that the above process is not computable if and only if x is a dyadic rational. And so we ask whether the Cauchy name of x is always 'close' to some dyadic rational, $(\exists n)(\exists m < n)(\forall k)(|m2^{-n} - x_k| < 2^{-k})$. Note that x_k is the kth rational in $x's$ Cauchy name. If no, complete the binary expansion construction above. If yes, check every n and $m \leq n$ until we find the pair n, m that satisfies the above sentence (which can be done with a Π_1^0 question). Then $x = \frac{m}{2^{-n}}$, which can be computably and finitely expressed in binary.

In both cases we obtain a binary expansion $x = 0.b_1 b_2 b_3 \dots$ which we will use to construct f.

Recall that we would like

$$f(x) = \begin{cases} t & \text{if } t \in [0,1] \text{ and } x - t^* \in D \\ 0 & \text{otherwise} \end{cases},$$

where $t \in [0,1]$ and t^*, D is defined as in Example 4.2.2.

We begin by defining a pairing function $\langle n, m \rangle$ as follows.

$$\langle n, m \rangle = \begin{cases} \Sigma_{i=1}^{n-1} i + m & \text{if } m \leq n \\ -1 & \text{otherwise} \end{cases}$$

This pairing function is meant to represent 'blocks' like those seen in the t^* function. We call a block a consecutive sequence of binary points that have been seen before in order, and one at the end that hasn't. i.e. $t^* = 0.t_1 t_1 t_2 t_1 t_2 t_3 \dots$, and so t_1 is the first block, $t_1 t_2$ the second block,

$t_1 t_2 \ldots t_k$ the kth block, etc. $\langle n, m \rangle$ gives the position of t_m in the nth block of t^*, assuming that $m > n$ (if this is not the case then we send the pairing function to -1 for completeness).

To define f we need to decide if at some stage x mimics t^* for some $t \in [0, 1]$. For simplicity assume the sth stage is when we observe b_s, the sth binary point in the approximation of x. We ask if there exists a stage s, and natural M, such that for all $m > M$, and $n \leq m$, we have $b_{s + \langle m, n \rangle} = b_{s + \langle m+1, n \rangle}$. If no, then there is no t such that $x - t^* \in D$, and so set $f(x) = 0$. If yes, we want to set $f(x) = t = 0.t_1 t_2 t_3 \ldots$. To define the kth binary point of t ask $(\exists s)(\exists M \geq k)(\forall m > M)(b_{s + \langle m, k \rangle} = 0)$. If yes, set $t_k = 0$, otherwise $t_k = 1$. Lastly, computably convert binary expression $0.t_1 t_2 t_3 \ldots$ into a Cauchy name for t. And so $f(x)$ outputs the Cauchy name for desired t (if t exists), and can be computed by $J^{(2)}$. $\square$

We have now shown that $J^{(2)}$ is sufficient to compute at least one canonical Darboux function. Applying a very similar argument to the construction in Theorem 4.2.3 now gives the following result.

Corollary 4.2.12. *If f is effective Barie class 2 then there exists a sequence of effective Baire class 2 functions $(f_i)_i$ with the Darboux property such that $\forall x f(x) = \lim_i f_i(x)$.*

4.2.3 Computational power of canonical Darboux functions

How can we discuss a real-valued function's Turing degree? If a function is continuous, this is not a problem; a function is continuous if and only if it is computable relative to some oracle.[3] However, what if the function in question is not continuous? This is an important issue to address, as canonical Darboux functions are by definition discontinuous, and we would like some way to discuss their complexity. Some work in this area

---

[3]Folklore or any early text in recursive analysis.

has been done, for example, see Scott ideals [51]. These fall outside of the scope of this thesis, and so we propose a different strategy. As every real is the limit of a sequence of rationals, we believe it is natural to discuss a function's complexity based on how it acts on the computable reals. If we take this approach, discussing the computational power of a function in some sense accesses the 'Markov computability' of that function. If nothing else, this notion ties back to the early chapters nicely, and so we relativise the Markov computability definition in order to discuss the computational power of a function f.

Definition 4.2.13. We call a function $f : \mathbb{R}_c \to \mathbb{R}_c$ *Markov computable in z* if z can compute the index function of f.

Notice that because we refer to $\mathbb{R}_c$, rather than to $\mathbb{R}$ relative to z, this is a *partial* relativisation of Markov computability. If a function f is Markov computable in z, and z computes y, we will sometimes informally refer to this situation as "f computes y".

Keeping this tentative definition in mind, we give a simple application. Define a simple discontinuous function $g : \mathbb{R}_c \to \mathbb{R}_c$ as follows.

$$g(x) = \begin{cases} 1 & \text{if } x < 0 \\ 0 & \text{if } x \geq 0 \end{cases}$$

Lemma 4.2.14. *If the function g (defined above) is Markov computable in z, then z is sufficient to compute $0'$.*

Proof: Fix e and suppose we want to know whether $\varphi_e(e)$ halts. We can decide this by building a computable real y which we will input into g such that $g(y) = 1$ if an only if $\varphi_e(e)$ halts.

We will built a computable Cauchy name $y_1, y_2, \ldots$ for our computable real y. At stage n, observe $\varphi_e(e)$ with use n at stage n. That is, ask does $\varphi_e(e)[n] \downarrow$? If yes, for all $m \geq n$ set $y_m = -2^{-(n+1)}$. We have now determined y. If no, set $y_n = 0$ and check $\varphi_e(e)$ with use $n + 1$ at stage $n + 1$.

If $\varphi_e(e)$ halts then we have built the Cauchy name of y such that $y < 0$, and so $g(y) = 1$. If, on the other hand, $\varphi_e(e)$ never halts, then $y_n = 0$ for all n, hence $g(y) = 0$. $\square$

The argument above can be applied to any discontinuous function, and leads us to conclude that every discontinuous function $f(x)$ can compute $0'$. Or more formally,

Lemma 4.2.15. *If a discontinuous function $f : \mathbb{R}_c \to \mathbb{R}_c$ is Markov computable in z, then z can compute $0'$.*

This result follows by generalising the argument given above. Given x (non-uniformly) at a discontinuity in f, the question becomes how to compute a sequence of computable reals approaching x that all map sufficiently far away from $f(x)$ to ensure discontinuity. We can apply the same argument as above once such a sequence has been found. The issue here is that, while we can observe an index that appears to output a computable real whose image is sufficiently far away from $f(x)$, it may turn out that this function is partial. We can avoid this by using the same trick as in Theorem 2.2.3.

Corollary 4.2.16. *At least $0'$ is required to compute every discontinuous function $f : \mathbb{R} \to \mathbb{R}$.*

We can now turn to the computational power of some of the earlier examples. Recall the function defined previously.

$$f(x) = \begin{cases} t & \text{if } t \in [0, 1] \text{ and } x - t^* \in D \\ 0 & \text{otherwise} \end{cases}$$

By restricting the domain and range of f to $\mathbb{R}_c$, we claim that, in the sense of definition 4.2.13, f can compute $0''$.

It is easier first to prove something similar to Lemma 4.2.14 explicitly for this function, and then extend the argument to achieve our result.

Theorem 4.2.17. *If the function f (defined above) is Markov computable in z, then z can compute $0'$.*

Proof: This result follows by a relatively straightforward construction.

Fix e. We will show that 'f can decide' whether or not $\varphi_e(e)$ halts. Then if z computes the index function of f, z must also compute $0'$. To do this, we construct a computable real y such that $f(y) = 0$ if and only if $\varphi_e(e) \uparrow$.

Consider the computable real $\gamma = 0.112123123412345\ldots$. Notice $\gamma = t^*$ if $t = 0.12345678\ldots$ (t^* as defined Example 4.2.2). Recall the definition of a 'block' given previously. We say the real x has entered a *block cycle* if after some stage it exactly mimics γ, i.e. every new term in x's Cauchy name differs from the former by an appropriate block. The y we want to build is almost γ, with a few changes according to the behaviour of $\varphi_e(e)$ at each stage.

Essentially, the function f can tell if at some point a real we input enters a block cycle. We want our real y to enter this block cycle if and only if $\varphi_e(e) \downarrow$.

To do this we observe $\varphi_e(e)$ at each stage. If $\varphi_e(e) \uparrow [n]$, we need to ensure the block cycle is broken in the nth term of y's Cauchy name, y_n. It is easiest to consider γ as the Cauchy sequence $\gamma_0 = 0$, $\gamma_1 = 0.1$, $\gamma_2 = 0.112$, $\gamma_3 = 0.112123$, $\gamma_n = 0.112123\ldots123\ldots n$. A new 'block' has been added to each consecutive term in this sequence, and $\lim_i \gamma_i = \gamma$. Similarly, at each stage we also add a block to the new term in the Cauchy name of y, *except* we add either $\mathbf{123}\ldots n$ or $\mathbf{023}\ldots n$ depending on the behaviour of $\varphi_e(e)[n]$.

Suppose we observe $\varphi_e(e) \uparrow [n]$. We then consider the last term y_{n-1} in the sequence we are building. Does $y_{n-1} = 0.\ldots\overbrace{\mathbf{123}\ldots(n-1)}^{\text{final block}}$ or $y_{n-1} = 0.\ldots\overbrace{\mathbf{023}\ldots(n-1)}^{\text{final block}}$? We need to ensure we break the (potential) block cycle. So, if the former, add $\mathbf{023}\ldots n$ to y_n, i.e. $y_n = 0.\ldots\mathbf{123}\ldots(n-$

$$\text{1) } \overbrace{\mathbf{0}23\ldots n}^{\text{new block}}. \text{ If the latter, add } \mathbf{1}23\ldots n, \text{ i.e. } y_n = 0.\ldots\overbrace{\mathbf{1}23\ldots n}^{\text{new block}}.$$

In this way, if for all n we have $\varphi_e(e) \uparrow$, the starting digit of each new block added to y_n will cycle between 0 and 1. This will build y so that there exists no $t \in [0,1]$ such that $y - t^* \in D$. Hence $f(y) = 0$.

If on the other hand there exists an n such that $\varphi_e(e) \downarrow [n]$, then from the stage that this becomes apparent we copy γ's Cauchy name exactly. In this case the block cycle has begun, and $f(y) = t$ (where $t = 0.12345\cdots \in [0,1]$ and because $y - t^* \in D$).

We have shown that f can decide whether or not $e \in 0'$. By extending this argument, for example following a similar pattern but altering every i between 0 and i to decide $i \in 0'$, we can conclude that f computes $0'$. Or more precisely, if z computes the index function of f, then z computes $0'$. $\square$

This argument can be adapted to show that 'f computes $0''$'. We will illustrate this now. For simplicity, we switch to using quantified predicates, rather than partial functions, in our argument.

Theorem 4.2.18. *If the function f (defined above) is Markov computable in z, then z can compute $0''$.*

Proof: Let $P(x_1, x_2, x_3)$ be a computable predicate, $x_1, x_2, x_3 \in \mathbb{N}$. We will show that the function f defined above can decide, for fixed e, the formula $\exists x_2 \forall x_3 P(e, x_2, x_3)$. This is done by again building a computable real y. We use an almost identical argument to that above, except we now require y to enter a block cycle if and only if $\exists x_2 \forall x_3 P(e, x_2, x_3)$. At stage $\langle b, c \rangle$ we check $P(e, b, c)$. The pairing function is explained in the construction.

Stage $\langle 1, 1 \rangle$: Check $P(e, 1, 1)$. If it returns true, set $y_1 = 0.1$. Then check $P(e, 1, 2)$. If $P(e, 1, 1)$ returns false, set $y_1 = 0.0$ and check $P(e, 2, 1)$ (that is, go to stage $\langle 2, 1 \rangle$).

Generally, if $P(e, 1, 1)$ returns true it is possible that $(\forall x_3)P(e, 1, x_3)$, therefore $\exists x_2 \forall x_3 P(e, b, c)$ is true. To confirm this, we continue checking $P(e, 1, x_3)$ until we observe a failure. If we never find a failure, it must be true that $\exists x_2 \forall x_3 P(e, x_2, x_3)$. On the other hand, if at any stage that we do observe a failure, we begin checking $(\forall x_3)P(e, 2, x_3)$. A failure is not enough to confirm that $\exists x_2 \forall x_3 P(e, x_2, x_3)$ is false. Only infinitely many failures would imply this. We need to build y to reflect this.

Stage $n = \langle b, c \rangle$: Suppose $P(e, b, x_3)$ retuned true for all $x_3 < c$ and $y_{n-1} = 0.1 \ldots q_m$. Technically, $m = \sum_{i=1}^{n-1} i$, and we assume here that we have observed $n - 1$ stages so far. Consider $x_3 = c$. If $P(e, b, c)$ returns true, add the block $123 \ldots n$ on to the end of y_{n-1} to form y_n. So $y_n = \overbrace{0.1 \ldots (n-1)}^{y_{n-1}} 123 \ldots n$. Now go to stage $\langle b, n+1 \rangle$ (i.e. check $P(e, b, n+1)$).

If $P(e, 1, n)$ returns false, let the current block in our new term y_n be different in the first digit to the last block in y_{n-1}. Hence, if $y_{n-1} = 0.1 \ldots \overbrace{123 \ldots (n-1)}^{\text{final block}}$, add the block $023 \ldots n$ on to the end of y_{n-1} to form y_n. That is, $y_n = \overbrace{0.1 \ldots (n-1)}^{y_{n-1}} \overbrace{023 \ldots n}^{\text{new block}}$. Similarly, if $y_{n-1} = 0.1 \ldots \overbrace{023 \ldots (n-1)}^{\text{final block}}$ then $y_n = \overbrace{0.1 \ldots (n-1)}^{y_{n-1}} \overbrace{123 \ldots n}^{\text{new block}}$. Now go to stage $\langle b+1, 1 \rangle$ (i.e. check $P(e, b+1, 1)$).

If there exists an b such that $(\forall c)\ P(e, b, c)$ returns true, then we will allow 1's in every block of y from some stage onwards, thereby achieving the block cycle. However, if no such c exists we can always find two consecutive blocks in y that begin with different digits, therefore never achieving a block cycle.

Just like above, $f(y) = 0$ if and only if $e \notin 0''$. By extending this argument to include all e, the result follows. $\square$

The results of this subsection apply to only a single example of a canonical Darboux function. It remains open whether or not this idea can be

extended somehow to all canonical Darboux functions.

4.3 The Darboux property on $\mathbb{Q}$

In this section, we shift our focus to a more classical version of computability. We could have included a discussion concerning functions defined on $\mathbb{R}_c$, however, because Markov computable functions are continuous on this domain, they have the Darboux property on $\mathbb{R}_c$ trivially. So we abandon $\mathbb{R}_c$ and instead focused on $\mathbb{Q}$. Perhaps we will be able to salvage some relationship between 'rational Darboux functions' and other notions of computability. In this section, we shake off the shackles of real computability (Type II and Markov) and allow our functions to be once again computable in the classical sense. We define the Darboux property on $\mathbb{Q}$ and discuss the difficulty of approximating a computable function $f : \mathbb{Q} \to \mathbb{Q}$ using these rational Darboux functions. In this section, unless otherwise stated, all results are original.

In this section only (unless otherwise stated) we will refer to a computable function $f : \mathbb{Q} \to \mathbb{Q}$ under a standard coding of $\mathbb{Q}$. We also use regular jump notation ($0'$, $0''$, etc.) with standard interpretation rather than the jump operator here.

Definition 4.3.1. We say a function $f : \mathbb{R} \to \mathbb{R}$ has the *rational Darboux property* if f is defined on $\mathbb{Q}$ and has the Darboux property when restricted to $\mathbb{Q}$. Namely, for all $a < b$, $a, b \in \mathbb{Q}$, and all $y \in [f(a), f(b)] \cap \mathbb{Q}$, we have $f(a), f(b) \in \mathbb{Q}$, and there exists $x \in [a, b] \cap \mathbb{Q}$ such that $f(x) = y$.

Call the class of rational functions with the rational Darboux property $\mathcal{RD}$.

We will restrict our discussion to explicit rational functions $f : \mathbb{Q} \to \mathbb{Q}$, but the arguments following could be extended to functions defined on all of $\mathbb{R}$. Notice that the rational Darboux property does not imply the real Darboux property (as given in Definition 4.1.1), nor does Definition 4.1.1

imply the rational Darboux property. These are two distinct notions.

4.3.1   The closure of the class $\mathcal{RD}$ is all rational functions

Our aim is to prove that, for any computable function $f : \mathbb{Q} \to \mathbb{Q}$, there exists a *computable* sequence of functions $(f_i)_i$ such that for all i, $f_i : \mathbb{Q} \to \mathbb{Q}$ has the rational Darboux property and $\lim_i f_i = f$.[4] We take a similar approach to the last section, first defining a canonical rational function, proving the principal result, and then effectivising it.

We define a rational canonical Darboux function and then give an example of such a function.

Definition 4.3.2. A function $f : \mathbb{Q} \to \mathbb{Q}$ is a *canonical rational Darboux function (C.R.D function)* if for every interval $I \subset \mathbb{Q}$, $f(I) = \mathbb{Q}$.

Example 4.3.3. [5] Let $\mathbb{D}$ be the set of dyadic numbers. Define a set of equivalence classes $\mathcal{A} = \{[x] : x \in \mathbb{Q}\}$ where $[x] = \{y \in \mathbb{Q} : x - y \in \mathbb{D}\}$ (i.e. $x \sim y \iff x - y \in \mathbb{D}$).

$\mathbb{D}$ is dense in $\mathbb{R}$ (and $\mathbb{Q}$), therefore $[x] = x + \mathbb{D} = \{x + d : d \in \mathbb{D}\}$.

Claim: $|\mathcal{A}| = |\mathbb{Q}|$.

Proof of claim: Clearly $|\mathcal{A}| \leq |\mathbb{Q}|$. In the other direction, suppose $|\mathcal{A}| \ngeq |\mathbb{Q}|$. Then $\mathcal{A}$ is finite, and we can list its members $[x_1], [x_2], \ldots, [x_n]$. The unique sequence each of its members must repeat after some stage determines each equivalence class. That is, for every member $x \in [x_i]$, there exists a $m \in \mathbb{N}$, such that after $x \upharpoonright m$, x repeats some fixed finite sequence. In doing so, we are identifying the dyadic rational that defines the equivalence class. But if we take one copy of each of the finite sequences corresponding to each of the classes in $\mathcal{A}$, placing them

---

[4]'Computable' is intended again in the classical sense here.

[5]This example is similar to an example of a real-valued canonical Darboux function given in [6].

together in any order, then we will obtain a new finite repeating sequence (i.e. dyadic rational) that is not represented in $\mathcal{A}$, a contradiction. $\square$

And so $|\mathcal{A}| = |\mathbb{Q}|$. Hence there exists a bijection $g : \mathcal{A} \to \mathbb{Q}$. Define a new function $f(x) = g([x])$.

Because $\mathbb{D}$ is dense in $\mathbb{Q}$, and dense in any rational open interval $(a, b) \cap \mathbb{Q}$, each equivalence class must have at least one element in (a, b). Therefore $f((a, b)) = g([a, b]) = g([\mathcal{A}]) = \mathbb{Q}$. $\square$

In the previous section, we gave a proof that every real-valued function is the limit of a sequence of Darboux functions. In parallel, we now prove that a similar result holds when we restrict our attention to $\mathbb{Q}$.

Theorem 4.3.4. *Every rational function* $f : \mathbb{Q} \to \mathbb{Q}$ *is the limit of a sequence of functions with the rational Darboux property.*

Proof: Let $f : \mathbb{Q} \to \mathbb{Q}$ be the function we are interested in approximating and let the equivalence class $\mathcal{A}$ be as defined above. Let $h : \mathbb{Q} \to \mathcal{A}$ be a bijection.

We want to split $\mathcal{A}$ up into countably many disjoint subsets. Define this partition as follows:

$\mathcal{A}_1 = h([0, 1))$

$\mathcal{A}_2 = h([-1, 0))$

$\mathcal{A}_3 = h([1, 2))$

$\dots$

$\mathcal{A}_k = h([\frac{-k}{2}, \frac{-k}{2} + 1))$, k even

$\mathcal{A}_{k+1} = h([\frac{k}{2}, \frac{k}{2} + 1))$

Notice $|\mathcal{A}_i| = |\mathbb{Q}|$ for all i; $|\mathcal{A}_i| \leq |\mathcal{A}| = |\mathbb{Q}|$ and $|\mathcal{A}_i| = |h[a, b]| = |[a, b) \cap \mathbb{Q}| \geq |(a, b) \cap \mathbb{Q}| = |\mathbb{Q}|$.

Therefore, there exists a bijection $m_i : \mathbb{Q} \to \mathcal{A}_i$ for all i.

Define a sequence of functions as follows.

$$
f_i(x) = \begin{cases} r & \text{if } x \in m_i(r) \\ f(x) & \text{otherwise} \end{cases}
$$

We partitioned $\mathcal{A}$ so that, for all i, j, we have $\mathcal{A}_i \cap \mathcal{A}_j = \emptyset$ and $\bigcup_i \mathcal{A}_i = \mathcal{A}$. Hence any $x \in \mathbb{Q}$ belongs to a unique class in $\mathcal{A}$, and so must belong to a unique class in exactly one $\mathcal{A}_i$.

Therefore, $\lim_i f_i(x) = f(x)$ for all $x \in \mathbb{Q}$.

All that remains is to show that each f_i has the rational Darboux property.

Consider any interval $I \subset \mathbb{Q}$. Each equivalence class is dense in $\mathbb{Q}$, which means $I \cap m_i(r) \neq \emptyset$ for all $r \in \mathbb{Q}$ and $i \in \mathbb{N}$. Therefore, there exists an $x \in I$ such that $x \in m_i(r) \cap I$, i.e. $f_i(x) = r$. This means $r \in f_i(I \cap m_i(r)) \subset f_i(I)$ for every $r \in \mathbb{Q}$, hence $f_i(I) = \mathbb{Q}$. $\square$

4.3.2   The complexity of canonical rational Darboux functions

Using Theorem 4.3.4 and Example 4.3.3 we now discuss the complexity of finding such a sequence for a given rational function $f : \mathbb{Q} \to \mathbb{Q}$. We first address the complexity of C.R.D. functions.

Lemma 4.3.5. *There exists a computable C.R.D. function* $f : \mathbb{Q} \to \mathbb{Q}$.

Proof: Example 4.3.3 is such a function.

Notice that we can computably decide whether two rationals x and y are in the same equivalence class. Assume that we are given two rationals $q_1 = \frac{a_1}{b_1}$ and $q_2 = \frac{a_2}{b_2}$ in reduced form. Without loss of generality, assume

$q_1 > q_2$. Subtraction is a computable operation, so compute $q_1 - q_2 = q_3 = \frac{a_3}{b_3}$. We ask whether or not q_3 is dyadic. That is, for every $n < b_3$ we ask if $b_3 = 2^n$. If no, then the q_1 and q_2 belong to different equivalence classes. If yes, then q_1 and q_2 differ by a dyadic, and so belong to the same class.

Now build a bijection $h : \mathbb{Q} \to \mathcal{A}$ in stages as follows. Enumerate $\mathbb{Q}$ and $\mathcal{A}$. We computably enumerate $\mathcal{A}$ by enumerating $\mathbb{Q}$, and for each rational q_n we observe, we check if it is in any of the classes that exist at this stage. There are only finitely many classes to check at any stage, so this process it computable. If q_n belongs to one of the classes that exist at the given stage, add q_n to that class. If not, define a new class $[q_n]$.

Suppose we observe q_s at stage s. Define $h(q_s)$ to be the 'oldest' equivalence class that has not yet been mapped to. If no classes are available at this stage, we wait until a new unseen class is enumerated into $\mathcal{A}$, and set $h(q_s)$ equal to this class.

It is clear that h is computable, and a bijection. Taking $g = h^{-1}$ gives us the bijection of Example 4.3.3 $\square$

Finally, we have the tools to give the main result of this section.

Theorem 4.3.6. *Every computable function $f : \mathbb{Q} \to \mathbb{Q}$ is the limit of a computable sequence of computable functions with the rational Darboux property.*

Proof: We need to prove that the function defined in Theorem 4.3.4, i.e.

$$f_i(x) = \begin{cases} r & \text{if } x \in m_i(r) \\ f(x) & \text{otherwise} \end{cases},$$

is computable for each i.

Take computable bijection $h : \mathbb{Q} \to \mathcal{A}$ given in Lemma 4.3.5 to be that used in Theorem 4.3.4.

We can computably find the partition of $\mathcal{A}$ any given equivalence class, $[q]$, is contained in. This is done by enumerating h on each of the unit intervals and waiting for $[q]$ to appear. We can detect this computably because deciding whether two rationals are in the same equivalence class is computable (see Lemma 4.3.5).

In this manner, each $\mathcal{A}_i$ is enumerable, and hence there exists a computable bijection m_i between $\mathbb{Q}$ and $\mathcal{A}_i$. The explicit construction is similar to that given in Lemma 4.3.5.

For any rational q, there exists exactly one $i \in \mathbb{N}$ and $r \in \mathbb{Q}$ such that $q \in m_i(r)$. Given i we cannot tell computably if there exists an r such that $q \in m_i(r)$, however we do know that such r and i exist.

To computably define the sequence we are interested in, we do not define functions individually, but rather we define all functions at once on a particular rational input. Suppose we are given $q \in \mathbb{Q}$. We define $f_i(q)$ for all i as follows.

Enumerate $\mathcal{A}_j$ for all $j \in \mathbb{N}$. At each stage ask if q is a member of any of the equivalence classes enumerated into each $\mathcal{A}_j$. The rational q is a member of some equivalence class, and so we will eventually find a class that contains it. Suppose $[q] \in \mathcal{A}_k$. We can computably find r such that $m_k(r) = [q]$ (again, enumeration). Set $f_k(q) = r$ and $f_i(q) = f(q)$ for all $i \in \mathbb{N} \backslash \{k\}$. $\square$

We can now deduce that the complexity of a rational function $f : \mathbb{Q} \to \mathbb{Q}$ entirely determines the complexity of its corresponding rational Darboux sequence.

Corollary 4.3.7. *If $f : \mathbb{Q} \to \mathbb{Q}$ and $f(x) \leq_T 0^{(n)}$ then there exists a sequence of rational Darboux functions $(f_i(x))_i$ such that f is the limit of $(f_i(x))_i$ and $(\forall x \in \mathbb{Q})(\forall i)(f_i(x) \leq_T 0^{(n)})$.*

4.4 Uniform limits of Darboux functions

In the previous sections, we explored real and rational valued functions and concluded that each could be approximated by a sequence of their respective (canonical) Darboux functions. Besides implicitly assuming a pointwise limit, we did not place any restrictions on the rate that these sequences approximated f. We now spend a moment discussing what happens if we require our Darboux sequences to *uniformly* approximate a function. In this section, we will restrict our attention to real-valued functions and the classical Darboux definition given earlier in the chapter.

The Bruckner, Ceder and Weiss paper mostly inspired this section [14]. Excluding the Examples 4.4.5 and 4.4.6, all work in this section is theirs.

4.4.1 Preliminaries

Sierpiński mentioned in [56] that, while every function is the limit of some sequence of Darboux functions, not every real-valued function is the *uniform* limit of such a sequence. He proved that the class of these functions was not trivial, and that there exist functions that are such limits and do not satisfy the Darboux property themselves. Bruckner, Ceder and Weiss classify this class of functions in [14]. We outline some of their argument here.

Recall a sequence of functions $f_1, f_2, \ldots$ is said to *uniformly converge* to a function f if, for all $\epsilon > 0$ there exists N such that $n > N$ implies $(\forall x \in \mathrm{dom}(f)) \; |f(x) - f_n(x)| < \epsilon$.

Definition 4.4.1. A function $f : I \to I$ is a member of the class $\mathcal{U}$ if, for every interval $[a, b] \subset I$ and every set A of cardinality less that c, the set $f([a, b] \backslash A)$ is dense in the interval $[f(a), f(b)]$.

Recall that if A and B are sets we say that A is *dense* in B if every open interval that intersects B also has nonempty intersection with A. We

say a set A is *c-dense* in B is every open interval that intersects B also contains c points of A.

In turns out that $\mathcal{U}$ is the uniform closure of the class of Darboux functions [14]. We give an outline of this result in the following subsection.

4.4.2   The uniform closure of $\mathcal{D}$

Before we sketch a proof of the theorem, note the following two results.

Theorem 4.4.2. *(Bruckner, Ceder, Weiss [14]) Let $f \in \mathcal{U}$ and $\epsilon > 0$. Then there exists a function $g \in \mathcal{U}$ such that g is not constant on any subinterval of its domain, the range of g is countable and $(\forall x \in dom(f))|f(x) - g(x)| < \epsilon$.*

Theorem 4.4.3. *(Bruckner, Ceder, Weiss [14]) Let $f \in \mathcal{U}$ such that f is not constant on any subinterval of its domain and the range of f is countable. Then f is the uniform limit of a sequence of Darboux functions.*

We now state their main result.

Theorem 4.4.4. *(Bruckner, Ceder, Weiss [14]) $f \in \mathcal{U}$ if and only if f is the uniform limit of a sequence of Darboux functions.*

Proof: The forwards direction follows by applying Theorems 4.4.2 and 4.4.3. In the other direction, let $f : I \to I$, $[a, b]$ be a closed subinterval of I and A a set with cardinality less that c. If we can prove that $f([a, b] \backslash A)$ is dense in $[f(a), f(b)]$ then we are done. Let U be any open interval whose closure is contained in $(f(a), f(b))$. We will show that $f([a, b] \backslash A) \cap U \neq \emptyset$.

Let $U = (y - \epsilon, y + \epsilon)$ and assume without loss of generality that $f(a) < f(b)$. There exists an n such that $(\forall x \in [a, b])|f_n(x) - f(x)| < \frac{\epsilon}{4}$ and such that $f_n(a) < y - \epsilon$ and $y + \epsilon < f_n(b)$. Then because f_n is Darboux it takes

every value in $(y-\epsilon, y+\epsilon)$ over $[a, b]$, hence there must exist a $x_0 \in [a, b] \backslash A$ such that $y - \frac{\epsilon}{2} < f_n(x_0) < y + \frac{\epsilon}{2}$. Therefore, $f([a, b] \backslash A) \cap U \neq \emptyset$, and the result follows. $\square$

4.4.3   Examples

We now give examples of functions that are not the uniform limit of any sequence of Darboux functions. Theorem 4.4.4 is a useful tool when constructing these.

First, notice that every Darboux function is, of course, a member of $\mathcal{U}$. We now give an effective Baire class 2 function that is not a member of $\mathcal{U}$.

Example 4.4.5. Define a function $f : \mathbb{R} \to \mathbb{R}$,

$$f(x) = \begin{cases} 1 & \text{if } x \in \mathbb{D} \\ 0 & \text{otherwise} \end{cases}.$$

The function $f(x)$ is computable in $J^{(2)}$ (we ask if x is dyadic, which we can decide with $J^{(2)}$ - for details see Theorem 4.2.11, construction of an interval function that can be computed in $J^{(2)}$). We can remove a countable set of points A in the interval $[0, \frac{1}{3}]$, namely all dyadic points, and $f([0, \frac{1}{3}] \backslash A)$ is not dense in $[f(0), f(\frac{1}{3})] = [1, 0]$. Therefore $f \notin \mathcal{U}$, and hence by Theorem 4.4.4 is not the uniform limit of any sequence of Darboux functions.

We can, in fact, modify this example to give an even simpler function $f \notin \mathcal{U}$, where this time f is effective Baire class 1.

Example 4.4.6. Define a function $f : \mathbb{R} \to \mathbb{R}$,

$$f(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}.$$

The function $f(x)$ is computable in J (to decide $x = 0$, given the Cauchy name $x_1, x_2, \ldots$ of x, we ask $\forall n \; |x_n - 0| < 2^{-n}$). We can remove a countable set of points A in the interval $[0, \frac{1}{3}]$, namely the point 0, and $f([0, \frac{1}{3}] \backslash \{0\}) = 0$ is not dense in $[f(0), f(\frac{1}{3})] = [1, 0]$. Therefore $f \notin \mathcal{U}$, and hence by Theorem 4.4.4 is not the uniform limit of any sequence of Darboux functions.

Chapter 5

Singular Points and Polynomials

5.1 Introduction

In this chapter, we decide how difficult it is to identify a set of special points, called singular points, for a polynomial of two variables. A singular point of a function $F(x, y)$ is a y-plane (by which we mean the collection of points $\{(x, y) : y = y_0 \in \mathbb{R}\}$ for some fixed y_0) that the function oscillates about in at least one direction. The formal definition is to follow, but we first provide some motivation for including this material. This chapter was largely inspired by the Bruckner, Ceder, and Weiss paper [14]. It turns out that continuous functions in the plane with a finite number of singular points have ties with the class $\mathcal{U}$, defined in the previous chapter. We state the relevant theorem here for interest.

Theorem 5.1.1. *(Bruckner, Ceder, Weiss [14]) Let $F(x, y)$ be continuous in the entire plane and have at most finitely many singular points. Then, if $f \in \mathcal{U}$ and g is continuous, the function h defined by $h(x) = F(f(x), g(x))$ is in $\mathcal{U}$.*

Discussing this result in detail falls outside of the scope of this thesis,

but the functions themselves are interesting enough to investigate further. We would like to decide how difficult it is to determine the set of singular points for a simple continuous function $F(x, y)$; the polynomial. It turns out that identifying these points is not as easy task. We also give constructive examples of functions with particular singular points. A similar strategy to the one used in this chapter could be used to determine how hard it is to find other characterised points for polynomials.

This section covers some of the discussions and results of [48], with application to a result in [14]. The proofs given for Lemmas 5.1.5, 5.2.2[1], 5.2.4, Corollary 5.2.3, and Examples 5.3.1, 5.3.2 and 5.3.3, are our own.

We give a non-standard definition of a singular point.

Definition 5.1.2. Let $F(x, y)$ be a function continuous on the entire plane. We call a point y_0 *singular* if one of the following does not exist.

$$\lim_{\substack{y \to y_0 \\ x \to \infty}} F(x, y) \text{ or } \lim_{\substack{y \to y_0 \\ x \to -\infty}} F(x, y)$$

Example 5.1.3. The function xy has a singular point at 0, while $x + y$, $\max(x, y)$ and $\min(x, y)$ have no singular points.

We can think of a singular point y_0 as a line in the plane that $F(x, y)$ oscillates about it in at least one direction. For those familiar with analysis and differential equations, note that our definition of a singular point is distinct from the standard definitions given in those courses. To begin with, when we talk about a singular 'point' here, we actually mean a line in the plane, where the standard definition truly refers to a *point* on the plane. It might seem strange to discuss an obscure 'point', but it turns out that the difficulty of calculating these points is linked to the problem of finding the roots of a polynomial. This is also an interesting question, and discussing singular points allows us to address both.

The following theorem can be found in [14]. We will revisit this result later on.

---

[1]The proof is our own, but this result is probably due to folklore.

Theorem 5.1.4. *(Bruckner, Ceder, Weiss [14]) A polynomial $P(x, y)$ has at most a finite number of singular points.*

Proof: Let

$$P(x, y) = \sum_{k=0}^{n} p_k(y)x^k,$$

where $p_k(y)$ is a *sub*-polynomial in y and $p_n(y) \neq 0$. We prove by induction on n that if $p_n(y_0) \neq 0$, then y_0 cannot be a singular point and $\lim_{\substack{y \to y_0 \\ x \to \pm\infty}} P(x, y) \neq 0$. If $n = 0$, every point y_0 is non-singular. Suppose the hypothesis holds for $n - 1$, and let $p_n(y_0) \neq 0$. Then

$$P(x, y) = p_0(y) + x \sum_{k=1}^{n} p_k(y)x^{k-1}.$$

By hypothesis,

$$\lim_{\substack{y \to y_0 \\ x \to \pm\infty}} \sum_{k=1}^{n} p_k(y)x^{k-1}$$

exists (as y_0 is not a singular point of $\sum_{k=1}^{n} p_k(y)x^{k-1}$), and is not equal to 0. Therefore, $P(x, y)$ tends to an infinite limit as y tends to y_0 and x to $+\infty$ and $-\infty$. As both limits given in the definition above exist, y_0 cannot be singular. Therefore, if y_0 is a singular point of $P(x, y)$ it must be a root of $p_n(y)$. $\square$

Extending the above argument, we can conclude the following.

Lemma 5.1.5. *If y_0 is a singular point of a polynomial $P(x, y) = \sum_{k=0}^{n} p_k(y)x^k$ then $p_j(y_0) = 0$ for all $0 < j \leq n$.*

Proof: We prove by induction on n that, for $0 \leq m < n$, if $p_{n-m}(y_0) \neq 0$ then y_0 is not singular and $\lim_{\substack{y \to y_0 \\ x \to \pm\infty}} P(x, y) \neq 0$. If $n = 0$, then every point y_0 is non-singular. Suppose the hypothesis holds for $n - 1$, and let $p_{n-m}(y_0) \neq 0$. Then

$$P(x, y) = p_0(y) + x \sum_{k=1}^{n} p_k(y)x^{k-1}.$$

By hypothesis,

$$\lim_{\substack{y \to y_0 \\ x \to \pm\infty}} \sum_{k=1}^{n} p_k(y)x^{k-1}$$

exists and is not equal to 0.

More explicitly, $\sum_{k=1}^{n} p_k(y)x^{k-1} = p_1(y)+p_2(y_0)x+\cdots+p_{n-m}(y)x^{n-(m+1)}+ \cdots + p_n(y)n^{n-1}$. By the inductive hypothesis, if the sub-polynomial $p_i(y_0)x^{n-(m+1)} \neq 0$ then y_0 cannot be singular. We assumed $p_{n-m}(y_0) \neq 0$, hence y_0 is not a singular point of $\sum_{k=1}^{n} p_k(y)x^{k-1}$. This, and the inductive hypothesis, then imply $\lim_{\substack{y \to y_0 \\ x \to \pm\infty}} \sum_{k=1}^{n} p_k(y_0)x^{k-1} \neq 0$. The result now follows by the same argument given in the theorem above. $\square$

## 5.2	Computing singular points

### 5.2.1	Building a set of potential singular points S

Let a polynomial $P(x,y) = \sum_{k=0}^{n} p_k(y)x^k$, where for all $0 < k \leq n$, $p_k(y) \neq 0$ is a polynomial with computable real coefficients. As above, we call $p_k(y)$ a *sub-polynomial* of $P(x,y)$. In this subsection, we would like to identify $P(x,y)$'s set of singular points. It turns out that we can at least computably construct a finite set S of *potential* singular points of $P(x,y)$. The true singular points will be a subset of this set, and will be harder to identify.

First note, as a consequence of Theorem 5.1.4, that y_0 is a singular point of $P(x,y)$ only if $p_n(y_0) = 0$ (we do not have the backwards implication here). Lemma 5.1.5 then extends this notion, and concludes y_0 is a singular point of $P(x,y)$ only if $p_n(y_0) = 0$ for all n (expect perhaps $n = 0$). Therefore, if we can compute the real roots of each of the n sub-polynomials (excluding $p_0(y)$) and take their intersection, we must obtain a set S that contains, at least, every singular point of $P(x,y)$. Since the set of roots of any polynomial is finite, S must also be finite.

The set S can, in fact, be constructed computably, providing that each sub-polynomial $p_i(y)$ is monic, and the coefficients of each sub-polynomial are members of $\mathbb{R}_c$. This result was originally presented by Pour-El and Richards in [53], reproved constructively by Chambers in [17], and published later in [48].

Theorem 5.2.1. *(Pour-El and Richards [53]) The real roots of a monic polynomial with computable real coefficients are computable.*

Pour-El and Richards gave a non-constructive proof that the exact roots of these types of polynomials are computable.[2] Unfortunately, their proof relies on the assumption that we can determine rationality and multiplicity. Chambers later gave a constructive proof of this result, which was published in [48]. It outlines an explicit algorithm to find these computable roots. We will not give the details here.

Consequently, the Chambers' algorithm almost gives us a computable set of potential singular points S. There is a caveat. Notice that Theorem 5.2.1 only claims its result for *monic* polynomials. It is easy to disregard this, as any polynomial can usually be expressed as a monic by dividing through by its leading coefficient. However, life is not so simple when we are concerned about computability! Our computable coefficients are given as rational Cauchy sequences, and while we can approximate each of them with arbitrary precision, we cannot computably decide whether what appears to be the leading coefficient is equal to 0, or not. This poses a problem if we need to divide by it.

Lemma 5.2.2. *If a polynomial P with computable real coefficients is not monic, then P's roots are not necessarily computable.*

Proof: Suppose there was a uniform procedure ψ taking every polynomial P to its roots. Consider the non-monic polynomial $P = 1 - bx$ for some $b \in \mathbb{R}_c \backslash \{0\}$. Let $b_1, b_2, \ldots$ be the Cauchy name of b. The root of P is $\frac{1}{b}$, and ψ outputs a Cauchy sequence r_i that converges rapidly to $\frac{1}{b}$. Suppose

---

[2]Theorems 8 and 9, pp 41-44.

the use of the mth rational approximation of root $\frac{1}{b}$ (i.e. r_m) is the nth term in the approximating sequence for b, and suppose it looks like it could have $b = 2^{-(n+1)}$ (which means $|b_n - 2^{-(n+1)}| < 2^{-n}$). For example, let the Cauchy name of b have $b_i = 2^{-(n+1)}$ for all $i \leq n$. Now to beat the machine. When ψ has committed $r_m \geq \frac{3}{4}$ (because $\frac{1}{b} = \frac{1}{2^{-(n+1)}} \geq 4$), we now claim $b = -2^{-(n+2)}$. We have not broken the approximation of b, the root of our polynomial is now negative, and ψ no longer outputs a Cauchy sequence that converges to the root of $P = 1 + 2^{-(n+2)}x$. Contradiction. $\square$

Of course, if we do not know that the relevant sub-polynomials are monic, we can decide this with the Π_1^0 question. Given the Cauchy name $q_1, q_2, \ldots$ of each leading coefficient, simply ask $(\forall n)|q_n - 0| < 2^{-n}$? Division is a computable operation, and, therefore, Theorem 5.2.1 allows us to conclude the following.

Corollary 5.2.3. *If $P(x, y) = \sum_{k=0}^{n} p_k(y)x^k$ is a polynomial with computable real coefficients then $0'$ is sufficient to determine a finite set which contains a subset of all singular points of $P(x, y)$.*

If, on the other hand, we do know that every sub-polynomial $p_k(y)x^k$ (excluding possibly $p_0(y)$) is monic, then we can computably find a finite set of potential singular points for a polynomial $P(x, y) = \sum_{k=0}^{n} p_k(y)x^k$.

Lemma 5.2.4. *If $P(x, y) = \sum_{k=0}^{n} p_k(y)x^k$ is a polynomial, $p_k(y)x^k$ for all $0 < k \leq n$ is monic, and each has computable real coefficients, then we can computably determine a finite set S which contains a subset of all singular points of $P(x, y)$.*

5.2.2   Refining S, the set of potential singular points

Refining this set poses a greater challenge. Recall that S is the intersection of roots for a polynomial $P(x, y)$, and let $S' \subseteq S$ be the true subset of singular points. If we want to identify S', then for every member y_0

of S we need to ask whether or not the limits

$$\lim_{\substack{y \to y_0 \\ x \to \pm\infty}} P(x,y)$$

exist. If a limit in one direction fails to exist, then y_0 is singular. However, this is a nontrivial question! For every $L \in \mathbb{R}$ we need to determine

$$(\exists \epsilon > 0)(\forall \delta, n)(\exists x, y \in \mathbb{R})(y \in B(y_0, \delta) \wedge (x > n) \to |P(x,y) - L| > \epsilon) \tag{5.1}$$

If the answer is 'yes' for all $L \in \mathbb{R}$, then y_0 is a singular point. Of course, we have some major complexity problems here, particularly that we are quantifying over all reals.

While we cannot easily construct our subset S' of singular points, we are able to refine S. Consider (1) for a particular limit L, taking values x, y over $\mathbb{Q}$ rather that $\mathbb{R}$,

$$(\exists \epsilon > 0)(\forall \delta, n)(\exists x, y \in \mathbb{Q})(y \in B(y_0, \delta) \wedge (x > n) \to |P(x,y) - L| > \epsilon) \tag{5.2}$$

Then if (2) is true, we know (1) must also be true (although no implication exists in the opposite direction of course). Therefore, we can potentially identify *some* members of S that are singular, although even then we need to use a relatively complex predicate.

This is as close as we get to identifying the actual set of singular points of a polynomial $P(x,y)$ with computable real coefficients. If every sub-polynomial (excluding $p_0(y)$) of $P(x,y)$ is monic, we can computably determine a finite set S of potential singular points. Therefore, in this case, we can computably determine the upper bound of singular points of P. If every sub-polynomial is not monic, then $0'$ is sufficient to rewrite the relevant sub-polynomials as monics, and a similar argument follows. In both cases, we may be able to further refine S, although even this is a much more complicated question, as we have just seen above.

5.3   Examples

We finish this chapter by giving a few constructive examples of different functions and singular points.

Example 5.3.1. *A continuous rational function that has a singular point at* 0.

Let $Q(x_1, x_2, x_3)$ be a Π_3^0 predicate. We will build a rational function $f : \mathbb{Q}^2 \to \mathbb{Q}$ such that:

1. if Q is false, then $\lim_{\substack{y \to 0 \\ x \to \pm\infty}} f(x, y) = 0$.

2. if Q is true, then 0 is a singular point.

This is a very similar argument to that given in Theorem 4.2.18

Let $Q(x_1, x_2, x_3) = \exists x_1 \forall x_2 \exists x_3 R(x_1, x_2, x_3)$, where R is a computable relation. Recall predicate (2) from above.

We now build a function f that will have a singular point at 0 if and only if Q is true. If Q is not true, we want $f(x, y)$ to approach 0 as y approaches 0 and x approaches $\pm\infty$. Let $\langle a, b, c \rangle$ represent the stage where we assess $P(a, b, c)$.

We begin giving the construction for x_1 at 1.

Stage $\langle 1, 1, 1 \rangle$: Check $R(1, 1, 1)$. The relation $R(1, 1, 1)$ is either true or false.

If $R(1, 1, 1)$ *is true:* choose large $u_0, v_0 \in \mathbb{Q}$ such that $v_0 \in B(0, 2^{-1})$ and $u_0 > 1$. Set $f(u_0, v_0) = 1$. We also enumerate two rationals q_1, q_2, and if $f(q_1, q_2)$ has not been defined, set it equal to 0. We will see later that setting $f(u_0, v_0) = 1$ breaks our continuity. So whenever this happens, we vary f in a cone like manner to avoid this: $(u_0, v_0, 1)$ is the cone point coordinate (i.e. $f(u_0, v_0) = 1$), the points at the rim of the cone take the value $f(x, y) = 0$, and all of the points in-between vary appropriately

between 0 and 1. Choose a small radius for this cone that avoids the point (q_1, q_2). Now go to stage $\langle 1, 2, 1 \rangle$.

If $R(1, 1, 1)$ is false: choose the same $u_0, v_0 \in \mathbb{Q}$ and set $f(u_0, v_0) = 0$. Enumerate two rationals q_1, q_2, and if $f(q_1, q_2)$ has not been defined, set it equal to 0. Now go to stage $\langle 1, 1, 2 \rangle$.

Notice that if $R(1, 1, 1)$ is true, it could be that $\forall x_2 \exists x_3 R(1, x_2, x_3)$. Therefore, we move to stage $\langle 1, 2, 1 \rangle$, and try our luck with $R(1, 2, 1)$. On the other hand, if $R(1, 1, 1)$ is false, there still could exist some x_3 such that $R(1, 1, x_3)$ is true, and so we check $R(1, 1, 2)$.

Stage $\langle 1, b, c \rangle$: Assume for all $x_2 < b$, we have found some x_3 such that $R(1, x_2, x_3)$ is true (this must be case if we have reached this stage). We now check $R(1, b, c)$.

If $R(1, b, c)$ is true: choose $u, v \in \mathbb{Q}$ such that $v \in B(0, 2^{-b})$, the rational $u > \max(b$, largest absolute value rational defined so far), and $f(u, v)$ has not yet been defined. Notice this will always be possible as $\mathbb{Q}$ is dense. Set $f(u, v) = 1$. Enumerate two more rationals into our list $q_1, q_2, \ldots$, and if $f(q_i, q_j)$, for all $i, j \leq n$ has not yet been defined, let $f(q_i, q_j) = 0$. Lastly, we need to ensure we maintain continuity. As in the first stage, we vary f in a cone like manner: (u, v) is the cone point coordinate, $f(u, v) = 1$, the points at the rim of the cone take the value $f(x, y) = 0$, and all of the points in-between vary appropriately between 0 and 1. As only finitely many points have been defined at this stage it is easy to avoid them. Choose a small distance d that does not interfere with any pre-defined points, and let $f(u \pm k, v \pm k)$, $k < d$ vary from 0 to 1 and 1 to 0 in a cone like manner. Go to stage $\langle 1, b+1, 1 \rangle$.

If $R(1, b, c)$ is false: choose the same u, v, but set $f(u, v) = 0$. Also enumerate two more rationals into our list $q_1, q_2, \ldots$, and if $f(q_i, q_j)$, for all $i, j \leq n$ has not yet been defined, let $f(q_i, q_j) = 0$. Go to stage $\langle 1, b, c+1 \rangle$.

This construction almost works. However, notice that we *also* need to

vary x_1; we want 0 to be a singular point if $\exists \mathbf{x_1} \forall x_2 \exists x_3 R(x_1, x_2, x_3)$. If we only ran the construction for $x_1 = 1$, it could return false, even though, for example, $\forall x_2 \exists x_3 R(2, x_2, x_3)$ is true. To ensure our construction works we will also dovetail over x_1. This means we will run one step for x_1 fixed at 1, i.e. $\langle 1, x_2, x_3 \rangle$. We then run two appropriate steps for x_1 set equal to 1 and 2, and three for x_1 set equal to 1, 2 and 3 and so on. In this way, if Q is false, after some point the function will be constant. However, if Q is true, then there are infinitely many stages where f is set not equal to 0, and this forces the function to oscillate (and therefore never attain a limit).

If Q really is true, we will observe infinitely many successes for some x_1, and so by the construction there will exist an x as large as we like and a y as close to 0 as we like such that $f(x, y) = 1$, i.e. $\lim_{\substack{y \to 0 \\ x \to \infty}} f(x, y) = 0$ does not exist. If, on the other hand, Q is false, for every x_1 after some stage we never observe another success. The construction then ensures that there exists an n such that, if $x, y > n$, $f(x, y) = 0$, i.e. $\lim_{\substack{y \to 0 \\ x \to \infty}} f(x, y) = 0$ exists. At each stage we maintained continuity, and so the function is well defined.

Example 5.3.2. *A continuous function where every rational point is singular.*

We build a new function $f : \mathbb{Q}^2 \to \mathbb{Q}$ that has a singular point at every rational by adapting the argument above. Let Q be a Π_3^0 predicate.

This time f is constructed such that:

1. if Q is false, then for all $q \in \mathbb{Q}$, $\lim_{\substack{y \to q \\ x \to \pm\infty}} f(x, y) = 0$.

2. if Q is true, then $\forall q \in \mathbb{Q}$, q is a singular point.

Let $\langle x, y, z \rangle$ be a pairing function as above. The argument is very similar to that above, so we omit some details. Let $|q|$ be the largest rational at stage $\langle a, b, c \rangle$ such that $\forall r > q$, $f(r, r)$ has not yet been defined.

If at stage $\langle a, b, c \rangle$ the relation $R(a, b, c)$ is true, choose u, v such that

$v > u > r$. Set $(\forall y)f(u, y) = 1$ and $(\forall y)f(v, y) = 0$. Vary $f(x, y)$ contin-
uously (i.e. saddle/linearly) between the line $f(u, y) = 1$ and $f(v, y) = 0$
on all values of y. We also need to ensure the function we are building
is continuous. At stage $(n - 1)$ the relation R returned either true or
false. In either case, there exists a greatest q such that $(\forall y)f(q, y) = 0$ or
$(\forall y)f(q, y) = 1$ (true and false respectively). Vary $f(x, y)$ continuously
between $f(q, y) = i$ and $f(u, y) = 1$ as appropriate to ensure continuity,
$i \in \{0, 1\}$.

If at stage $\langle a, b, c \rangle$ the relation $R(a, b, c)$ is false, choose v such that $v > r$
and set $(\forall y)f(v, y) = 0$. Ensure continuity of function as in the case
above.

The result follows.

Example 5.3.3. *A real-valued function where every real point is singular.*

An example would be a canonical Darboux function. Although, in-
evitably, we lose continuity.

Chapter 6

The Blaschke Selection Theorem

6.1 Introduction

The Blaschke Selection Theorem was originally proved by Blaschke, an Austrian differential and integral geometer, in 1916.[1] Recall that a set $X \subseteq \mathbb{R}^n$ is said to be *convex* if for any two points $x_1, x_2 \in X$, all points of the form $ax_1 + bx_2$, where $a, b \geq 0$ and $a + b = 1$, also belong to X. The Blaschke Selection Theorem asserts that every infinite collection of closed convex sets of a bounded portion of $\mathbb{R}^n$ contains a subsequence that converges to a nonempty closed convex subset [5]. In some sense, the Blaschke Selection Theorem is a generalisation of the Bolzano-Weierstrass Theorem, a fundamental result from classical analysis. The Bolzano-Weierstrass Theorem states that any infinite collection of *points* in a bounded portion of $\mathbb{R}^n$ contains a convergent subsequence. Proved in 1817 by Bolzano, the Bolzano-Weierstrass theorem was originally a lemma for a proof of the Intermediate value Theorem [7].[2] It was later

---

[1]The paper cited here, 'Kreis und Kugel', is written in German and has not been translated into English. A number of secondary sources confirmed this reference, for example [36] and [29].

[2]An English translation of this paper can be found in [9].

re-proved by Weierstrass and recognised as significant in its own right.[3] Similar to the Bolzano-Weierstrass Theorem, the Blaschke Selection Theorem has great practical importance, and so it is a useful theorem to effectivise. Also, as far as we know, the Blaschke Selection Theorem has not yet been considered from a computability theoretic perspective. In this chapter, we give a proof of the theorem, followed by a discussion of its complexity. We would like to establish how difficult it is to find, and compute the limit of, a convergent subsequence of an infinite collection of computable closed convex sets in a bounded portion of $\mathbb{R}^n$.

The main result of this chapter is as follows.

Theorem 6.3.8 *Every infinite computable collection of closed convex sets of a bounded portion of $\mathbb{R}^n$ has a convergent subsequence with a $0''$ computable limit.*

We also give a proof that $0''$ in Theorem 6.3.7 cannot be replaced by $0'$.

Because convexity is an important property, we finish the chapter by looking into the complexity of convexity. We prove that $0'$ is not sufficient to decide the convexity of a closed set in $\mathbb{R}$ (a result which can be generalised to $\mathbb{R}^n$), but that the set of indices of closed convex sets is co-c.e. over $0'$. Theorems 6.3.7-9, 6.4.1-3, Lemmas 6.3.2, 6.3.6, and Example 6.3.1 are new. The proof of Theorem 6.2.5 is original (although the result is attributed to Blaschke [5]).

6.2   The Blaschke Selection Theorem

Let us begin by defining the metric we will be working in. Let B be a bounded portion of $\mathbb{R}^n$ and $\mathcal{C}$ be the collection of closed sets contained in B. Recall, the distance from set a $A \subset \mathbb{R}^n$ to a point x is $d_A(x) = \inf\{|x - y| : y \in A\}$.

---

[3]Weierstrass published relatively little in his lifetime, but some of his original works can be found in [65]. See [25] for historical information.

Definition 6.2.1. Let X be a closed set in $\mathbb{R}^n$ and take $\delta > 0$. Define $B(X, \delta)$ to be $\{x : d_X(x) < \delta\}$.

We define a metric in $\mathcal{C}$.[4] Let $X_1, X_2 \in \mathcal{C}$, and let δ_1 be the infimum of the collection of distances δ such that $X_2 \subset B(X_1, \delta)$, and δ_2 be the infimum of the collection δ' such that $X_1 \subset B(X_2, \delta')$. We define the distance between X_1 and X_2 to be

$$\Delta(X_1, X_2) = \delta_1 + \delta_2.$$

Δ is a metric. The proof is trivial, although we mention that two sets X_1, X_2 need to be closed in order for $\Delta(X_1, X_2) = 0$ to imply $X_1 = X_2$.

Definition 6.2.2. A sequence of closed sets (X_i) *converges* to limit X if $\lim_i \Delta(X_i, X) = 0$ and $X \in \mathcal{C}$.

We will use the following results in our proof of the Blaschke Selection Theorem. The first can be found in [24].

Lemma 6.2.3. *(Eggleston [24]) Let (X_i) be a convergent subsequence of elements of $\mathcal{C}$ with limit X. If every member of (X_i) is convex then X is also convex.*

Proof: Suppose that X is not convex. Then there exist two members x_1 and x_2 of X such that the line segment connecting x_1 and x_2 contains a point x_0 and $x_0 \notin X$. But X is closed, and so there must exist a $\delta > 0$ such that $B(x_0, \delta) \cap X = \emptyset$.

The sequence (X_i) converges to X, therefore there exists an i such that $\Delta(X_i, X) < \frac{1}{4}\delta$. Notice this means $X \subset B(X_i, \frac{1}{4}\delta)$ and $X_i \subset B(X, \frac{1}{4}\delta)$ and allows us to find $u \in X_i$ such that $|x_0 - u| < \frac{1}{4}\delta$. We do this by choosing members u_1 and u_2 of X_i within $\frac{1}{4}\delta$ of x_1 and x_2 respectively. There must exist a point on the line segment connecting u_1 and u_2 that

---

is within $\frac{1}{4}\delta$ of x_0, and this point must be a member of X_i by convexity. By the same argument, there exists a $v \in X$ such that $|u - v| < \frac{1}{4}\delta$.

This implies that there exists a point $v \in X$ such that $|x_0 - v| < \frac{1}{2}\delta$, a contradiction. $\square$

Lemma 6.2.4. *Let (X_i) be a nested decreasing sequence of closed sets in $\mathcal{C}$. Then $\bigcap_i X_i = X$ is the limit of this sequence.*

Proof: Suppose this is not the case. Then there exists a δ such that for infinitely many i, $\Delta(X_i, X) > \delta$. Take an element x_i from each X_i such that $d_X(x_i) > \delta$, and form a sequence $(x_i)_i$.[5] Let x be an accumulation point of this sequence. $x \notin X$ by assumption, but because each set X_i is closed, and our sequence is nested, x must be a member of each X_i, hence $x \in X$. Contradiction! $\square$

We now prove the Blaschke Selection Theorem. Our intention is to give a proof that is easy to analyse from a computability theoretic perspective.

Theorem 6.2.5. *(Blaschke [5]) (The Blaschke Selection Theorem) Every infinite collection of closed convex subsets of a bounded portion of $\mathbb{R}^n$ contains an infinite subsequence that converges to a closed nonempty convex subset of this bounded portion of $\mathbb{R}^n$.*

Proof: We work in $\mathbb{R}^3$ for simplicity. Let B be a bounded portion of $\mathbb{R}^3$ that contains an infinite collection $\mathcal{A}$ of closed convex subsets. By compactness, we can finitely cover B with open balls of radius $\epsilon > 0$.

We will first explain the construction for this fixed cover.

In the construction, we want to define a collection M_i of closed refinements of B, and $\mathcal{N}_i$ infinite refinements of $\mathcal{A}$. At the end of stage 1 we will obtain two new sets, $B_1 = \bigcap_i M_i$ and $\mathcal{A}_1 = \bigcap_i \mathcal{N}_i$. We will then choose a set from $\mathcal{A}_1$ and begin forming our convergent subsequence.

---

[5] Notice we are using two different metrics here. Recall that d was defined earlier; $d_X(x_i) = \inf_{x \in X} |x - x_i|$.

Let the fixed cover of B require n balls of radius ϵ. We first explain how to construct the collections $M_1, \ldots, M_n$ and $\mathcal{N}_1, \ldots, \mathcal{N}_n$.

Initially, take an open ball b_1 in our cover and ask whether infinitely many sets in $\mathcal{A}$ fail to intersect b_1 (asking whether there are infinitely many sets in $\mathcal{A}$ that are contained in $B \backslash b_1$ would be the same question phrased slightly differently).

If this is the case, define a new set $M_1 = B \backslash b_1$ and new subset of closed convex sets $\mathcal{N}_1$, where $\mathcal{N}_1$ is all closed convex sets from the original collection $\mathcal{A}$ that *do not* intersect b_1; $\mathcal{N}_1 = \{X : X \in \mathcal{A} \text{ and } X \cap b_1 = \emptyset\}$.

If not, let $M_1 = B$ and $\mathcal{N}_1 = \mathcal{A}$.

Notice that $\mathcal{N}_1$ remains an infinite collection of closed convex subsets of our space in both cases and M_1 is a closed bounded (and compact) subset of B.

Take another open ball in our cover b_2 and ask whether infinitely many sets that remain in $\mathcal{N}_1$ are members of $M_1 \backslash b_2$. If yes, define $M_2 = M_1 \backslash b_2$ and the new subset of closed convex sets $\mathcal{N}_2 = \{X : X \in \mathcal{N}_1 \text{ and } X \cap b_2 = \emptyset\}$. If no, let $M_2 = M_1$ and $\mathcal{N}_2 = \mathcal{N}_1$.

Assuming we have just defined M_i and $\mathcal{N}_i$, take the open ball b_{i+1} in our cover and ask whether infinitely many sets that remain in $\mathcal{N}_i$ are members of $M_i \backslash b_{i+1}$. If yes, define $M_{i+1} = M_i \backslash b_{i+1}$ and the new subset of closed convex sets $\mathcal{N}_{i+1} = \{X : X \in \mathcal{N}_i \text{ and } X \cap b_{i+1} = \emptyset\}$. If no, let $M_{i+1} = M_i$ and $\mathcal{N}_{i+1} = \mathcal{N}_i$.

Once this algorithm has been completed for each b_i in our cover, we have formed two collections $M_1, \ldots, M_n$ and $\mathcal{N}_1, \ldots, \mathcal{N}_n$. Notice that each collection is nested, and if $X \in \mathcal{N}_i$ then $X \subset M_i$. In addition, the construction has also ensured that if b_j and b_k are two balls in our cover such that $b_j \cap M_n \neq \emptyset$ and $b_k \cap M_n \neq \emptyset$ (i.e. neither ball was explicitly removed in the construction), then finitely many closed convex sets in $\mathcal{N}_n$ *do not intersect both b_j and b_k*. This follows because b_j was not removed from M_{j-1} to form M_j, which means only finitely many closed convex

sets in the refined collection $\mathcal{N}_{j-1}$ *did not* intersect b_j (if we had removed b_j we would have left only finitely many closed convex sets in the new collection N_j, something we want to avoid). Similarly for b_k.

This condition applies to every pair of balls that were not removed in the construction. Therefore, it follows that we can eliminate finitely many closed convex sets from $\mathcal{N}_n$ to form a new set $\mathcal{N}_{n+1}$ such that, if $X \in \mathcal{N}_{n+1}$, then it intersects *every* ball in the cover that was not explicitly removed in the construction.

Finally, we let $B_1 = \bigcap_i^n M_i$ and $\mathcal{A}_1 = \bigcap_i^{n+1} \mathcal{N}_i$. Choose any set that remains in $\mathcal{A}_1$, call it C_1. This will be the first term in our convergent subsequence.

Before moving on, we note the following facts:

1. B_1 is a closed subset of B because $M_1 \supset \cdots \supset M_n$ and each M_i was a closed subset of B.

2. $\mathcal{A}_1$ is an infinite subset of the original collection of closed convex sets $\mathcal{A}$ because $\mathcal{N}_i \supset \cdots \supset \mathcal{N}_{n+1}$ and each $\mathcal{N}_i$ was an infinite subset of $\mathcal{A}$.

3. If $X \in \mathcal{A}_1$ then $X \subseteq B_1$. By contrapositive, if $X \in \mathcal{A}$ and $X \nsubseteq B_1$ then there exists a least i such that $X \nsubseteq M_i$, but $X \subseteq M_{i-1}$. By construction there is a ball from the cover b_i such that $M_i = M_{i-1} \backslash b_i$, hence $X \cap b_{i+1} \neq \emptyset$. But also by construction, $\mathcal{N}_i = \{X : X \in \mathcal{N}_{i-1} \text{ and } X \cap b_{i+1} = \emptyset\}$. Therefore, $X \notin \mathcal{N}_i$ and so $X \notin \mathcal{A}_1$.

4. If $X \in \mathcal{A}_1$ then $\Delta(X, B_1) < 2\epsilon$. This follows because if $X \in \mathcal{A}_1$, then X intersects every open ball in the cover that also intersects B_1 (i.e. those that were not removed). Therefore, as $X \subset B_1$ and $B \subset B(X, 2\epsilon)$, we have that $\Delta(X, B_1) < 2\epsilon$.

5. If $X_1, X_2 \in \mathcal{A}_1$ then $\Delta(X_1, X_2) < 4\epsilon$. This follows by the triangle inequality. $\Delta(X_1, X_2) \leq \Delta(X_1, B_1) + \Delta(X_2, B_1) < 2\epsilon + 2\epsilon$.

Now repeat with a cover of open balls of radius $\frac{\epsilon}{2}$, beginning with the bounded space B_1 and infinite collection of closed convex sets $\mathcal{A}_1$.

Continuing this construction, we obtain a sequence of closed convex sets $C_1, C_2, \dots$ and a decreasing sequence of closed sets $B_1 \supset B_2 \supset \dots$. Let $C = \bigcap_i B_i$.

Claim: $\lim_i C_i = C$

Proof: Again, we begin by listing a number of facts:

a. $C \neq \emptyset$ because it is a nested intersection of nonempty compact sets (recall that each B_i is compact).

b. $\forall n,\ C_n \subseteq B_n$ and $m > n$ implies $C_m \subseteq B_n$. This is because $C_n, C_{n+1}, \dots \in \mathcal{A}_n$, and if $X \in \mathcal{A}_n$ then $X \subseteq B_n$.

c. $\forall \delta\ \exists i$ such that if $X_1, X_2 \in \mathcal{A}_i$ then $\Delta(X_1, X_2) < \delta$. Take the cover $\epsilon = \frac{\delta}{4}$. The result then follows by an extension of Fact 4 above.

d. In addition, by Facts 4 and c if $X \in \mathcal{A}_i$ then $\Delta(X, B_i) < \delta$.

e. $\forall \delta\ \exists k\ \forall j > k$ such that $\Delta(B_j, C) < \delta$ by Lemma 6.2.4.

The result now follows because for all δ, there exists an i such that for all $j > i$ we have $\Delta(C_j, B_j) < \delta$. By Lemma 6.2.4 $\lim_i B_i = C$, so it follows that $\lim_i C_i = C$.

We have now constructed a convergent subsequence (C_i) of $\mathcal{A}$ such that $\lim_i C_i = C$. C is closed and nonempty as it is the infinite decreasing intersection of closed compact sets. Finally, C is convex by Lemma 6.2.3. $\square$

6.3 Subsequence and limit complexity

In the previous section, we gave a proof of the Blaschke Selection Theorem. Given a collection of sets, we will now discuss how difficult it is to

identify a convergent subsequence and compute its limit. We begin by restricting our attention to sequences of points in the unit interval, and then extend our argument to $\mathbb{R}^n$.

6.3.1   Restricting to the unit interval

Naturally, we begin by asking whether every computable subsequence of a collection of convex sets needs to converge to a computable limit? Unsurprisingly, this turns out to be false. We give an example in the unit interval.

Example 6.3.1. We define a halting sequence (in binary) as follows. Let the ith term in the sequence have a 1 in the ith binary point if and only if $\varphi_j(j)[i] \downarrow$. Otherwise, let this binary point be a 0. Each term in this sequence is computable, and the sequence converges to a limit x. However, if x was computable we could compute the halting problem. $\square$

Setting our sights lower, and still considering only the unit interval, we *can* show is that $0''$ is sufficient to find and compute the limit of a subsequence of any infinite collection of *points*.[6] In the following lemma, we effectivise the Bolzano-Weierstrass Theorem.

Lemma 6.3.2. *Given a computable infinite bounded sequence in $\mathbb{R}$, $0''$ is sufficient to compute a fast convergent subsequence, and hence compute its limit.*

Proof: Without loss of generality, assume the sequence $(x_i)_i$ is contained in the unit interval. We know that at least one of the following sets contains infinitely many elements.

$$\{x_i : 0 \leq x_i < \frac{1}{2}\}, \{x_i : \frac{1}{2} < x_i \leq 1\}, \{x_i : x_i = \frac{1}{2}\}$$

We can decide, for example, if the first set $X = \{x_i : 0 \leq x_i < \frac{1}{2}\}$ is infinite by asking if $(\forall n)(\exists m > n)(\exists s)(x_{n,s} \in [2^{-s}, \frac{1}{2} - 2^{-s}))$, where $x_{n,s}$ is

---

[6]The jump is considered as a set here again.

the sth term in the Cauchy name of x_n.[7] Similarly for the other two sets. Once we have identified one of the infinite sets, we extract an arbitrary element from it. Call this element y_1. If $y_1 = \frac{1}{2}$ we are done. Otherwise, without loss of generality, assume $y_1 \in \{x_i : 0 \le x_i < \frac{1}{2}\}$. Now repeat the process for the three sets:

$$\{x_i : 0 \le x_i < \frac{1}{4}\}, \{x_i : \frac{1}{4} < x_i \le \frac{1}{2}\}, \{x_i : x_i = \frac{1}{4}\}.$$

Again, one of which must contain infinitely many elements.

Continuing in this manner, we obtain a subsequence $y_1, y_2, \ldots$. By construction, this sequence is Cauchy and therefore, converges to a point in the unit interval. $\square$

To determine whether this result also holds in general we must discuss how we need to adapt our argument to work in higher dimensions.

6.3.2 Extending to higher dimensions

To give a generalised version of Lemma 6.3.2, we need to recall what we mean by a 'computable collection of sets'. In an earlier section we defined a closed set X to be computable if it had a computable distance function d_X. We will call a collection of closed sets (X_i) computable if there exists an algorithm that uniformly gives the distance function of term X_i for all i.

Also, Lemma 6.3.2 states that $0''$ is sufficient to approximate the limit of a sequence of points, which means $0''$ can compute a Cauchy name for this limit. In order to generalise this notion, we define what it means for a collection of sets to approximate another.

Definition 6.3.3. A sequence of closed sets (X_i) *approximates* a set X

---

[7] Notice that x_n is thought of as a real here, rather than a term in a Cauchy name for a real x. This is one of the only notational exceptions of this type, and was made for convenience.

if $\forall \epsilon \ \exists i \ \forall j > i \ \Delta(X_j, X) < \epsilon.$[8]

We can now relativise this definition and the definition of a computable sequence of closed sets.

Definition 6.3.4. A sequence of closed sets (X_i) *y-approximates* a set X if $\forall \epsilon \ \exists i \ \forall j > i \ \Delta(X_j, X) < \epsilon$ and y can compute such an i for any ϵ.

Definition 6.3.5. A collection of closed sets (X_i) is *y-computable* if there exists an algorithm computable in y that uniformly gives the distance function of set X_i for all i.

We give a lemma that will be useful later.

Lemma 6.3.6. *If a sequence of closed sets (X_i) is y-computable, and (X_i) y-approximates a set X, then X is y-computable.*

Proof: We need to show that y is sufficient to compute the distance function of X. That is, for any point x_0, we will show y is sufficient to give a suitable n^{th} term in a Cauchy name of $d_X(x_0)$. The sequence (X_i) y-approximates the set X, so y is sufficient to compute an i such that for all $j > i$, $\Delta(X_j, X) < 2^{-(n+1)}$. Taking any $j > i$, y is sufficient to compute the distance function of X_j, and hence can compute $d_{X_j}(x_0)$ to within $2^{-(n+1)}$. Let this rational approximation be q.

Claim: q is a sufficient n^{th} term in a Cauchy name of $d_X(x_0)$.

The claim follows by two kinds of triangle inequality. However, as d_X is not actually a metric, this may not be immediately apparent. We first show that

$$d_X(x) \leq d_X(z) + d_z(x) \tag{6.1}$$

$$d_x(z) \leq d_X(z) + d_X(x) \tag{6.2}$$

holds for all closed, bounded sets X and points x and z.

[8]This is just a more convenient reformulation of Definition 6.2.2.

Equation (6.1):

$$
\begin{aligned}
d_X(z) + d_z(x) &= \inf_{t \in X} |t - z| + \inf_{t \in y} |t - x| \\
&= \inf_{t \in X} |t - z| + |z - x| \\
&= \min_{t \in X} |t - z| + |z - x| \text{ as } X \text{ compact} \\
&= |w - z| + |z - x| \text{ for some } w \in X \\
&\geq |w - x| \text{ as } || \text{ a metric} \\
&\geq \inf_{t \in X} |u - x| \\
&= d_X(x)
\end{aligned}
$$

Equation (6.2):

$$
\begin{aligned}
d_X(z) + d_X(x) &= \inf_{t \in X} |t - z| + \inf_{t \in X} |t - x| \\
&= \min_{t \in X} |t - z| + \min_{t \in X} |t - x| \text{ as } X \text{ compact} \\
&= |w_1 - z| + |w_2 - x| \text{ for some } w_1, w_2 \in X \\
&\geq |w_1 - z| + |w_1 - x| \text{ w.l.o.g. choosing } w_1 \\
&\geq |z - x| \text{ as } || \text{ a metric} \\
&= d_x(z)
\end{aligned}
$$

Proof of claim: $q - 2^{-(n+1)} < d_{X_j}(x_0) < q + 2^{-(n+1)}$. Therefore, there exists a point $u \in X_j$ such that $q - 2^{-(n+1)} < |u - x_0| < q + 2^{-(n+1)}$. We are non-uniformly provided with x_0, so we can check every computable point in X_j until we find one that satisfies $q - 2^{-(n+1)} < |u - x_0| < q + 2^{-(n+1)}$. In addition, $\Delta(X_j, X) < 2^{-(n+1)}$, therefore every point $x \in X_j$ is within $2^{-(n+1)}$ of some point in X, i.e. $d_X(u) < 2^{-(n+1)}$. These details are summarised in Figure 6.1.
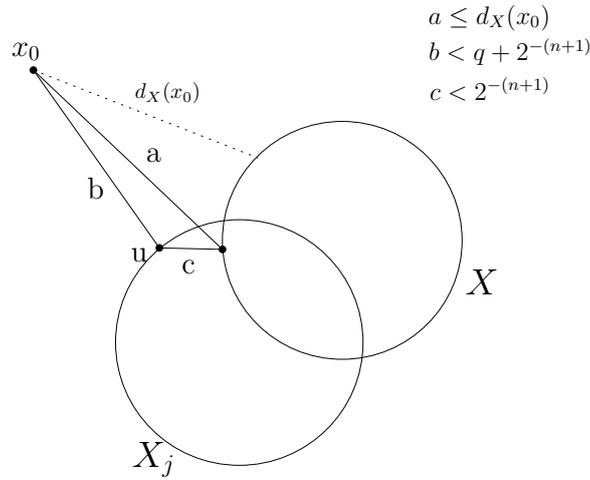
Figure 6.1: Distances between x_0, u, X, and X_j.

We now apply equation (6.1),

$$
\begin{aligned}
d_X(x_0) &\leq d_X(u) + d_u(x_0) \\
&= d_X(u) + |u - x_0| \\
&< 2^{-(n+1)} + q + 2^{-(n+1)} \\
&< 2^{-n} + q
\end{aligned}
$$

and equation (6.2),

$$
\begin{aligned}
d_X(x_0) &\geq d_{x_0}(z) - d_X(z) \\
&> q - 2^{-(n+1)} - 2^{-(n+1)} \\
&= q - 2^{-n}
\end{aligned}
$$

Hence, q is a sufficient n^{th} approximation of $d_X(x_0)$ and y is sufficient to compute q. The result follows. $\square$

We can finally prove that Lemma 6.3.2 holds in general. We give a proof below using details from Theorem 6.2.5 (the Blaschke Selection Theorem).

Theorem 6.3.7. *Given a computable, infinite collection of computable closed convex subsets of a bounded portion of $\mathbb{R}^n$, $0''$ is sufficient to find a convergent subsequence of this collection and approximate its limit.*

Proof: This result follows by evaluating the complexity of the sequence (C_i) constructed in Theorem 6.2.5, albeit making one small change. Notice that we cannot computably decide for a closed set X and an open set Y whether $X \cap Y \neq \emptyset$. So rather than asking at each stage whether the open ball b_i in a cover does not intersect infinitely many of our convex sets (as in the original proof), we will instead use the closure of b_i in the open ball's place. The original proof remains unchanged, but we can now computably decide whether $X \cap \overline{b_i} \neq \emptyset$, which will be useful below.

We call the closed set X_i, at stage t, the set $X_{i,t}$ (it is nice to think of these sets in Baire space).[9] Let (X_i) be an infinite, bounded, computable collection of closed convex sets. Finding a finite cover of our bounded portion of $\mathbb{R}^n$ is computable. We would now like to ask if there are infinitely many sets X_i in our current collection $\mathcal{N}_i$ (as defined in Theorem 6.2.5) that do not intersect b_i - or equivalently do not intersect $\overline{b_i}$. That is, $(\forall r)(\exists s > r)(\exists t)(X_{s,r} \in \mathcal{N}_{i,r}$ and $X_{s,t} \cap \overline{b_i} = 0)$?

Depending on the answer, we take the appropriate step as outlined in the construction. Note that $\mathcal{N}_i$ is c.e. and so $\mathcal{N}_{i,t}$ ($\mathcal{N}_i$ at stage t) is computable. The final question is asked when constructing $\mathcal{N}_{n+1}$. For each $X_s \in \mathcal{N}_i$, we ask whether $(\forall i, t)$ $((b_i$ in cover$) \wedge (\overline{b_i} \cap M_{i,t} \neq \emptyset))$ implies $(\overline{b_i} \cap X_{s,t} \neq \emptyset)$. If this is the case, enumerate X into $\mathcal{N}_{i+1}$. Otherwise, discard X. Note that the sequence (C_i) approximates its limit by definition. The result follows. $\square$

---

[9]If a closed set X has a computable distance function then it can be expressed as a Π_1^0 class in Baire space. Check $d_X(x)$. If the nth approximation of $d_X(x)$ is ever greater than 2^{-n}, kill the branch that extends however much of the Cauchy name of x we have seen.

Notice that, while proving both Lemma 6.3.2 and 6.3.7, the sequences we constructed also happened to approximate their limits. As a consequence of our construction, this is not always the case.

It now follows by this Theorem, and Lemma 6.3.6, that $0''$ is sufficient to compute the distance function of the limit X given above, which ties this result back to the original notion of a computable set (Chapter 4).

Theorem 6.3.8. *Every computable, infinite collection of closed convex subsets of a bounded portion of $\mathbb{R}^n$, has a convergent subsequence with a $0''$ computable limit.*

Finally, we note that Theorem 6.3.8 can, of course, be relativised. In all likelihood our result is equivalent to arithmetic comprehension over RCA_0 [57]. Unfortunately there was insufficient time to confirm this.

6.3.3   Back to the unit interval; $0'$ is not sufficient

While $0''$ is sufficient to find a convergent subsequence and approximate its limit, we note that $0'$ is not. We sketch the proof of this for a single dimension.

Theorem 6.3.9. *There exists a computable sequence $(x_i)_i$ contained in the unit interval such that no accumulation point of the sequence is computable in $0'$.*

Sketch of Proof: We want to build a sequence of points $x_0, x_1, \ldots$ in the unit interval such that no $0'$ computable sequence converges to any accumulation point in this sequence. We will first discuss how to build such a sequence that avoids a single $0'$ computable sequence, and then generalise this notion to account for *every* $0'$ computable sequence.

Let $y_0, y_1, \ldots$ be a $0'$ computable sequence. This means we are not truly 'given' the terms in the sequence $y_0, y_1, \ldots$ but rather $0'$ computable

Cauchy names converging to each of these terms. $y_{0,0}, y_{0,1}, y_{0,2} \ldots$ 'converging' to, for example, y_0 and so on. We do not know the rate at which each of these 'subsequences' converges, only that eventually they must settle on a value y_i.

At stage 0, we observe $y_{0,0}, y_{1,0}, y_{2,0} \ldots$ and assume at this point that $y_0, y_1, \ldots$ actually converges (of course, it may not, in which case we do not need to worry about avoiding it anyway). We now split the unit interval into two halves and consider $y_{0,0}$. Whichever half $y_{0,0}$ is in, we choose the first term in our sequence to be a point from the other half of $[0, 1]$. For example, if $y_{0,0} = \frac{5}{8}$, we could choose $x_0 = \frac{1}{4}$. We now observe $y_{1,1}$. Whichever half $y_{1,1}$ is in, we choose x_1 to be in the other. For example, perhaps $y_{1,1} = \frac{1}{4}$, then we could choose $x_1 = \frac{3}{4}$ (if $y_{1,1}$ is not in the same half as x_0, we simply set $x_1 = x_0$). Continuing this process, we know that the sequence $y_{0,n}, y_{1,n}, \ldots$ must eventually get close to y_n. Therefore, if $y_0, y_1, \ldots$ actually converges, avoiding the sequence $y_{0,0}, y_{1,1}, y_{2,2}, \ldots, y_{n,n}, \ldots$ will be sufficient to avoid it too. If $y_0, y_1, \ldots$ does not converge, then $(x_i)_i$ may not be able to avoid it. But if the sequence does not converge, then it cannot compute an accumulation point anyway.

So we have built a sequence $(x_i)_i$ that has no accumulation points computable by the $0'$ sequence $y_0, y_1, \ldots$.

We will now extend this argument by describing how to, in addition to the sequence above, avoid a second $0'$ computable sequence $w_0, w_1, \ldots$. At stage 0 we now consider both $y_{0,0}$ and $w_{0,0}$. We can computably decide whether $y_{0,0} \in [0, \frac{1}{2})$ or $\in [\frac{1}{2}, 1]$. We then ask which quarter $w_{0,0}$ is a member of: $w_{0,0} \in [0, \frac{1}{4}), [\frac{1}{4}, \frac{1}{2}), [\frac{1}{2}, \frac{3}{4})$ or $[\frac{3}{4}, 1]$. Choose a point for x_0 that avoids both the half and quarter which $y_{0,0}$ and $w_{0,0}$ are members of respectively. Continuing on in this manner (next checking $y_{1,1}$ and $w_{1,1}$), we can build a sequence $(x_i)_i$ that avoids the sequences $(y_i)_i$ and $(w_i)_i$ if they converge.

We can generalise this notion and build a sequence whose accumulation points avoid *every* $0'$ sequence. Take a computable listing of every $0'$

sequence, and extending the argument above, avoid certain terms in every sequence appropriately. For example, at stage n we would address the first n sequences on our list. Observe the nth approximation of each of the nth terms in these n sequences, and avoid them by choosing a point x_n that avoids the half, quarter, eighth, ..., $\frac{1}{2^{-n}}$th that the respective terms fall in.

We can see by the construction that there will always be an interval 'free' at each stage, so we can always define a new x_i. By dovetailing, our sequence $(x_i)_i$ will avoid the limit point (if it exists) of every $0'$ sequence. $\square$

Because $0'$ is not sufficient to compute any of $(x_i)_i$'s accumulation points, it follows that no subsequence of $(x_i)_i$ can converge to a $0'$ computable limit.

In general, this result holds in higher dimensions, as $\mathbb{R}$ embeds into $\mathbb{R}^n$.

6.4 Discussions on convexity

We now take a moment to explore the notion of convexity, and how hard it is to decide if a closed set A in $\mathbb{R}^n$ is convex. We first prove the following result for closed sets in Cantor space (as Π_1^0 classes).

Theorem 6.4.1. $0'$ *is not sufficient to decide the convexity of a closed set A in Cantor space.*

Proof: Suppose, to the contrary, that $0'$ was sufficient to determine convexity. Let A be a Π_1^0 class in Cantor space that $0'$ claims is convex. That is, we begin enumerating A, and at stage s, observing A_s, $0'$ declares that A is convex. Call a branch in A_s *active* if it has not permanently halted earlier in the enumeration. Otherwise, we say that this branched is *dead* or was *killed* at an earlier stage.

Choose an appropriate active branch in A_s such that *killing* this branch
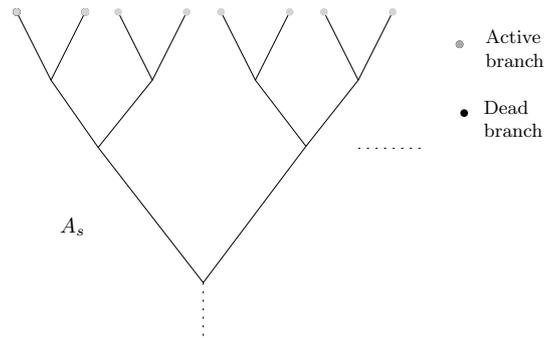
Figure 6.2: Tree at stage s.

will prevent A from being convex. It must be an active branch with at least one active branch to the left and right of it. We then declare this branch dead. We have now removed an open ball entirely contained in A_s so that A_{s+1} threatens to be non-convex (it has a hole!). Figure 6.2 shows A at stage s and Figure 6.3 A at stage $s+1$.
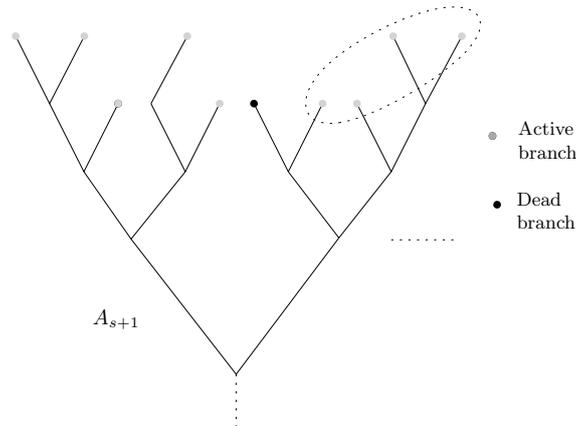
Figure 6.3: Tree at stage $s+1$.

Now suppose that $0'$ observes A_{s+1}, and states that A is now not convex. We then choose an appropriate collection of branches to kill in order to ensure that A_{s+2} again looks convex. Figure 6.4 shows A at stage $s+2$.

$0'$ must now claim that A is convex.

Repeating this method builds a singleton $\{x\} = A$, but $0'$ cannot deter-
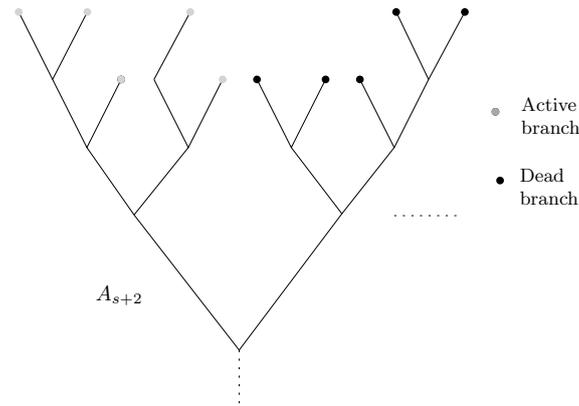
Figure 6.4: Tree at stage $s + 2$.

mine its convexity. Contradiction. $\square$

Because closed sets in Cantor space embed into closed sets in $\mathbb{R}$ (they are effectively homeomorphic), showing that $0'$ is insufficient to decide the convexity of a closed set in Cantor space is also a counterexample for $\mathbb{R}$. Therefore, we can conclude the following.

Theorem 6.4.2. *$0'$ is not sufficient to decide the convexity of a closed set A in $\mathbb{R}$.*

We can generalise this notion to $\mathbb{R}^n$ by interpreting a Π_1^0 class as a closed set in $\mathbb{R}^n$, and giving a similar argument.

The next thing we ask is how much *more* computational power is required to determine complexity? It turns out not very much more.

In this proof we explicitly discuss effectively closed sets in $\mathbb{R}^n$. Recall that an effectively closed set in $\mathbb{R}^n$ can be thought of as follows; at stage 0 it begins as all of $\mathbb{R}^n$, and at certain stages open balls of rational radius drop of out of the set.

Theorem 6.4.3. *The set of indices of effectively closed, convex sets of $\mathbb{R}^n$ is co-c.e. over $0'$.*

Proof: Suppose A is an effectively closed subset of $\mathbb{R}^n$. The set A fails to

be convex when an open ball falls out of A, and there exist two points in A such that the line segment connecting them has a non-empty intersection with the open ball that was removed. See Figure 6.5.
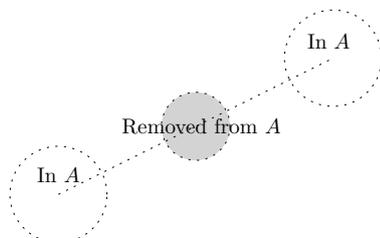

Figure 6.5: If A not convex.

We claim that $0'$ is sufficient to eventually determine if A is not convex. Let B be an open ball with rational center point and radius such that $B \cap A_{s-1} \neq \emptyset$, but $B \cap A_s = \emptyset$, where A_s is the set A at stage s. Finitely cover the space A_s within s of the ball B with open balls whose center points are rational, and of some appropriate small rational radius $\epsilon \in \mathbb{Q}$. For each pair of balls B_i and B_j in this cover, ask if for all $x \in \overline{B_i}$ and $y \in \overline{B_j}$, the line segment connecting x and y intersects B. This is decidable in $0'$. See Figure 6.6. Enumerate every such pair of balls into a set X.
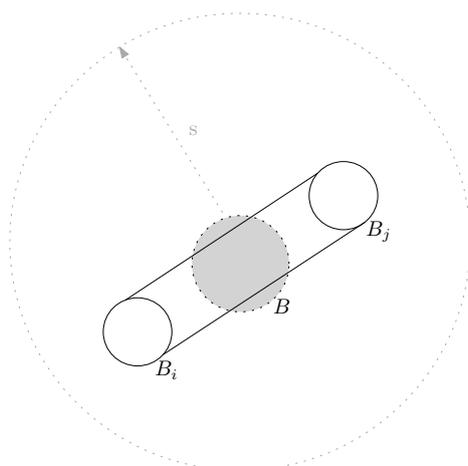

Figure 6.6: Does B threaten A's convexity?

Notice that $0'$ can also decide if any particular ball $\overline{B_i}$ in this cover at some stage falls out of A. This is because $\overline{B_i}$ is closed, and has so itself has a finite open cover (by compactness). The only way $\overline{B_i}$ has empty intersection with A is if, at some stage t, every open ball in some cover of $\overline{B_i}$ falls out of A. That is, $\overline{B_i} \not\subseteq A$ if and only if there exists a stage t such that $\overline{B_i} \not\subseteq A_t$. By the same principal, $0'$ can also decide if any particular pair of balls $\overline{B_i}, \overline{B_j}$ both fall out of A. That is, we ask if there exists a stage t such that $\overline{B_i} \not\subseteq A_t$ and $\overline{B_j} \not\subseteq A_t$.

Now take a pair of balls B_i and B_j that were enumerated (as a pair) into X. A fails to be convex if both B_i and B_j are truly subsets of A. So we ask if there exists a stage t such that $\overline{B_i} \not\subseteq A_t$ and $\overline{B_j} \not\subseteq A_t$. If no, then A fails to be convex. If yes, we cannot conclude anything (as A may fail to be convex for a different pair of balls, or different removed set B).

We repeat this construction for every open ball removed from A, and dovetail (increasing s cover) to ensure that, if A is convex, at some point we will come across a pair of balls that observe this. $0'$ was sufficiently complex to compute all of our questions, and the result follows. $\square$

Further Questions

There are a number of further questions that time did not allow us to answer. We list a few of them here.

1. Is it true that every real-valued function is the limit of a sequence of strictly non-canonical Darboux functions? What is the maximum complexity of a canonical Darboux function? We have shown that there at least exist canonical Darboux functions that are computable via the double jump function.

2. Can rational functions with the Darboux property somehow approximate real-valued functions?

3. How complicated is the class $\mathcal{U}$ (the class of real-valued functions that are the uniform limits of a sequence of Darboux functions)? There is a result that states that any Baire class 1 function in $\mathcal{U}$ satisfies the Darboux property [14]. How can we effectivise the definition of $\mathcal{U}$, and still have that any Baire class 1 function that is a member of $\mathcal{U}$ also has the Darboux property?

4. Recalling the notation used in the definition of $\mathcal{U}$ (see Definition 4.4.1). If f is Baire class 1, how complex must the set A be so that $f \notin \mathcal{U}$? How hard is it to find an interval $[a, b]$ that the function fails on.

5. Can we improve the complexity bound in the effective Blaschke Selection Theorem (Theorem 6.3.7)?

List of Figures

Bibliography

[1] O. Aberth. *Computable analysis.* McGraw-Hill International Book Company, 1980.

[2] O. Aberth. *Computable calculus.* Academic Press, 2001.

[3] J. Avigad and V. Brattka. Computability and analysis: the legacy of Alan Turing. pages 1–46, 2012.

[4] S. Banach and S. Mazur. Sur les fonctions calculables. *Ann. Soc. Pol. de Math*, 16:223, 1937.

[5] W. Blaschke. *Kreis und Kugel.* Veit, Berlin, 1916.

[6] B. Bogoel. Functions with the intermediate value property. *Romanian Mathematical Gazette Series A*, 1-2, 2012.

[7] B. Bolzano. *Rein analytischer beweis des Lehrsatzes, dass zwischen je zwey werthen, die ein entgegengesetztes resultat gewähren, wenigstens eine reelle wurzel der gleichung liege.* Gottlieb Haase Sohne Publisher, Prague, reprint: 2 edition, 1817.

[8] B. Bolzano. Spisy Bernarda Bolzano - Bernard Bolzano's Schriften Vol. 1. In K. Rychlik, editor, *Functionenlehre*. Královská Ceská Spolecnost Nauk, Prague, 1930.

[9] B. Bolzano and S. Russ (translator). Purely Analytic Proof of the Theorem that between any two Values, which give Results of Opposite Sign, there lies at least one real Root of the Equation. In *The*

mathematical works of Bernard Bolzano. Oxford University Press, Oxford, 2004.

[10] E. Borel. Le calcul des intégrales définies. *Journal de Mathéatiques pures et appliquées*, 6(8):159–210, 1912.

[11] V. Brattka, P. Hertling, and K. Weihrauch. A tutorial on computable analysis. *New Comp. Parad.*, pages 425–491, 2008.

[12] M. Braverman and M. Yampolsky. *Computability of Julia sets.* Springer-Verlag Berlin Heidelberg, 2009.

[13] L. Brouwer. *Collected works, Vol I - Philosophy and foundations of mathematics.* American Elsevier Publishing Company, Amsterdam, 1975.

[14] A. Bruckner, J. Ceder, and M. Weiss. Uniform limits of Darboux functions. *Coll. Math.*, 15:65–77, 1966.

[15] J. Caldwell and M. Pour-El. On a simple definition of computable function of a real variable-with applications to functions of a complex variable. *Z. Math. Logik Grundlagen Math.*, 21:1–19, 1975.

[16] G. Ceitin. Algorithmic operators in constructive metric spaces. *Tr. Math. Inst. Steklov*, 67:295–361, 1962.

[17] S. Chambers. *Constructing exact roots for polynomials: towards a constructive solution of the computable eigenvalue problem.* Masters thesis, Manchester University, 1997.

[18] J. Conway. Explanation given at Canada/USA Mathcamp. 2013.

[19] G. Darboux. Mémoire sur les fonctions discontinues. *Ann. Sci. Scuola Norm*, 4:57–122., 1875.

[20] M. Dehn. Transformation der Kurven auf zweiseitigen Flächen. *Math. Ann.*, 72(3):413–421, 1912.

[21] M. Dehn. J. Stillwell (translator). In *Papers on group theory and topology*, pages 179–199. Springer-Verlag New York Inc., 1987.

[22] O. Demuth. Necessary and sufficient conditions for Riemann integrability of constructive functions. *Dokl. Akad. Nauk SSSR*, 176:757–758, 1967.

[23] R. Downey, editor. *Turing's Legacy: Developments from Turing's ideas in logic.* Cambridge University Press, 2014.

[24] H. Eggleston. *Convexity.* Cambridge University Press, Cambridge, 1977.

[25] Encyclopedia.com. Weierstrass, Karl Theodor Wilhelm., 2008.

[26] K. Gödel. On formally undecidable propositions of Principia Mathematica and related systems I (1931). In *Collected Works, Vol. 1: Publications 19291936*, pages 144–195. Oxford University Press, New York and Oxford, 1986.

[27] R. Goodstein. *Recursive analysis.* Dover Publications (reprint 2010), Amsterdam, 1961.

[28] R. Gordon. *The integrals of Lebesgue, Denjoy, Perron, and Henstock (vol. 4).* Amer. Math. Soc., 1994.

[29] P. Gruber and J. Wills, editors. *Handbook of convex geometry Volume A.* Elsevier Science Publishers B.V., Amsterdam, 1993.

[30] A. Grzegorczyk. Computable functionals. *Fund. Math.*, 42:168–202, 1955.

[31] J. Heinonen. Lectures on Lipschitz analysis. pages 1–77, Jyvskäyä Summer School, 2004.

[32] G. Hermann. Die frage der endlich vielen schritte in der theorie der polynomideale. *Math. Ann.*, 95:736–788, 1926.

[33] G. Hermann. The question of finitely many steps in polynomial ideal theory (translation). *ACM SIGSAM Bulletin*, 32(3):8–30, 1998.

[34] P. Hertling. Banach-Mazur computable functions on metric spaces. In J. Blanck, V. Brattka, and P. Hertling, editors, *Computability and*

Complexity in Analysis, volume 2064, pages 69–81. Springer Berlin Heidelberg, 2001.

[35] P. Hertling. A Banach-Mazur computable but not Markov computable function on the computable real numbers. *Annals of Pure and Applied Logic*, 132(2-3):227–246, 2005.

[36] A. Ivanov. Blaschke selection theorem, 2014.

[37] S. Kleene. *Introduction to metamathematics*. Ishi Press (reprint 2009), 1952.

[38] S. Kleene. *Countable functionals*. North-Holland Publishing Company, Amsterdam, 1959.

[39] G. Kreisel. Review of "Meschkowski - Zur rekursiven funktionentheorie, Acta Math., 95, 1956, 9-23". *Math. Reviews 19*, 238, 1958.

[40] G. Kreisel, D. Lacombe, and J. Shoenfield. Partial recursive functionals and effective operations. In A. Heyting, editor, *Construc. in Math.*, pages 290–297, Amsterdam, 1959. North-Holland Publishing Company.

[41] L. Kronecker. Grundzüge einer arithmetischen theorie der algebraischen grossen. *J. Reine Angew. Math.*, 92:1–123, 1882.

[42] L. Kronecker, R. Dedekind, and J. Molk. *Grundzuge einer arithmetischen theorie der algebraischen grossen: a complete translation-except only abstracts of the first four sections.* California Institute of Technology, Pasadena, Calif, 1900.

[43] B. Kushner. *Lectures on constructive mathematical analysis*, volume 60. American Mathematical Society, 1980.

[44] R. Kuyper and S. Terwijn. Effective genericity and differentiability. *Journal of Logic & Analysis*, 6(4):1–14, 2014.

[45] D. Lacombe. Extension de la notion de fonction récursive aux fonctions d'une ou plusieurs variables éelles II and III. *C.R. Acad. Sci*, 241:13–14, 151–153, 1955.

[46] D. Lacombe. Extension de la notion de fonction récursive aux fonctions d'une ou plusieurs variables réelles I. *C.R. Acad. Sci*, 240:2478–2480, 1955.

[47] D. Lacombe. Les ensembles récursivement ouverts ou fermés, et leurs applications á l'analyse récursive. *C. R. Acad. Sci. Paris*, 245(13):1040–1043, 1957.

[48] D. Lester, S. Chambers, and H. Lee Lu. A constructive algorithm for finding the exact roots of polynomials with computable real coefficients. *Theoretical Comp. Sci.*, 279(1-2):51–64, 2002.

[49] A. Markov. On the continuity of constructive functions. *Uspehi Mat. Nauk*, 9:226–230, 1954.

[50] A. Markov. On constructive functions. *Trudy Math. Inst. Steklov*, 52:315–348, 1958.

[51] J. Miller. Degrees of unsolvability of continuous functions. *Journal of Sym. Logic*, 69(2):555–584, 2004.

[52] V. Orevkov. A constructive map of the square into itself which moves every constructive point. *Dokl. Akad. Nauk SSSR*, 152:55–58, 1963.

[53] M. Pour-El and J. Richards. *Computability in analysis and physics*. Springer, Heidelberg, 1989.

[54] D. Radcliffe. A function that is surjective on every interval. *Math. Ass. of Amer.*, 123(1):1–2, 2016.

[55] H. Rice. Recursive real numbers. *Proc. Amer. Math. Soc.*, 5:784–791, 1954.

[56] W. Sierpinski. Sur une propriété de fonctions réelles quelconques définie dans les espaces métriques. *Le Matematiche (Catania)*, 8:73–78, 1953.

[57] S. Simpson. *Subsystems of second order arithmetic*. Cambridge University Press, New York, 2009.

[58] E. Specker. Nicht konstruktiv beweisbare satze der analysis. *Journal of Sym. Logic*, 14:145–158, 1949.

[59] E. Specker. Der satz vom maximum in der rekursiven analysis. *Construc. in Math.*, Proc. Coll:254–265, 1959.

[60] A. Turing. On computable numbers, with an application to the Entsheidungsproblem. *Proc. of the London Math. Sc.*, 42:230–265, 1936.

[61] A. Turing. On computable numbers, with an application to the Entscheidungsproblem. A correction. *Proc. of the London Math. Sc.*, 43(2):544–546, 1937.

[62] A. Van Rooij and W. Schikhof. *A second course on real functions.* Cambridge University Press, 1982.

[63] R. von Mises. Grundlagen der wahrscheinlichkeitsrechnung. *Math. Z.*, 5:52–99, 1919.

[64] R. von Mises. *Richard von Mises, probability, statistics, and truth.* Dover Publications (reprint, original Springer: 1928), New York, 1957.

[65] K. Weierstrass. *Original Works. Weierstrass' writings were published as Mathematische Werke. 7 vols.* Berlin. 1894-1927.

[66] I. Zaslavsky. The refutation of some theorems of classical analysis in constructive analysis. *Uspehi Mat. Nauk*, 10:209–210, 1955.

[67] I. Zaslavsky. Some properties of constructive real numbers and constructive functions. *Trudy Math. Inst. Steklov*, 67:385–457, 1962.