On Bar Recursive Interpretations of Analysis

A Dissertation
Submitted in Partial Fulfillment
of the Requirements of the
Degree of Doctor of Philosophy,
Queen Mary University of London

By

Thomas Rhidian John Powell

August 2013

Abstract

This dissertation concerns the computational interpretation of analysis via proof interpretations, and examines the variants of bar recursion that have been used to interpret the axiom of choice. It consists of an applied and a theoretical component.

The applied part contains a series of case studies which address the issue of understanding the meaning and behaviour of bar recursive programs extracted from proofs in analysis. Taking as a starting point recent work of Escardó and Oliva on the product of selection functions, solutions to Gödel's functional interpretation of several well known theorems of mathematics are given, and the semantics of the extracted programs described. In particular, new game-theoretic computational interpretations are found for weak König's lemma for Σ^0_1-trees and for the minimal-bad-sequence argument.

On the theoretical side several new definability results which relate various modes of bar recursion are established. First, a hierarchy of fragments of system T based on finite bar recursion are defined, and it is shown that these fragments are in one-to-one correspondence with the usual fragments based on primitive recursion. Secondly, it is shown that the so called 'special' variant of Spector's bar recursion actually defines the general one. Finally, it is proved that modified bar recursion (in the form of the implicitly controlled product of selection functions), open recursion, update recursion and the Berardi-Bezem-Coquand realizer for countable choice are all primitive recursively equivalent in the model of continuous functionals.

Contents

Acknowledgements

First and foremost I am deeply grateful to my supervisors Paulo Oliva and Edmund Robinson for their academic guidance, patience and support during my time as a research student, without which this dissertation would not have been possible.

I am indebted to many people for stimulating mathematical discussions, in particular Martín Escardó, Ulrich Berger and Monika Seisenberger, who also invited me to visit them. It was a great honour to have been examined by Max Kanovich and Ulrich Kohlenbach, and I thank them for their many helpful comments.

Finally, I thank the EPSRC for funding my research for the last three and a half years.

Preface

The last century has seen a sweeping change in the central goals of metamathematics. Starting with Hilbert's metamathematical program of the 1930s, which was broadly concerned with providing a finitary justification for modern mathematical methods, logic and proof theory have experienced a gradual shift of emphasis away from foundational problems and towards *applications* in mainstream mathematics and computer science. Today this new direction is embodied by research programs such as the proof mining program, or the formalisation of mathematics in theorem provers.

In line with this shift of emphasis is a change in the way that we view traditional metamathematical tools and techniques. Good examples of this phenomenon are proof interpretations such as Gödel's functional interpretation: originally developed to establish relative consistency results, they have since been reoriented to extract constructive information from non-constructive proofs, leading in particular to new and improved results in several areas of mathematics.

In this dissertation we study bar recursive functional interpretations of analysis. We examine the meaning and behaviour of bar recursive programs extracted from proofs, and clarify the relationship between the different variants of bar recursion. Our modest aim is to acquire a better understanding of proof interpretations and their associated modes of recursion, but in doing so we hope to take a small step towards making metamathematical methods more accessible and applicable in mathematics itself.

CHAPTER ONE

Introduction

Proof interpretations are powerful tools that allow us to make computational sense of non-constructive ideas in mathematics. They have their origin in Hilbert's program where they were initially developed in order to obtain relative consistency proofs. However, in the 1950s Kreisel advocated the use of proof interpretations to systematically extract realizers for existential theorems, through 'unwinding' computational information hidden in the logical structure of their proofs. It is this feature that forms the basis of most modern applications of proof interpretations, although their significance in this respect has only become fully appreciated since the 1990s and the advent of the proof mining program, in which refinements of Gödel's functional interpretation have been applied to obtain new quantitative results in several areas of mathematics, including numerical analysis and ergodic theory.

Despite these achievements, in many ways we still understand surprisingly little about the action of proof interpretations and what they tell us about the semantics of classical proofs. Realizers extracted from all but the most trivial proofs are highly complex - their meaning often obscured beneath a heavy layer of syntax. This is particularly true when analysing proofs that involve choice or comprehension principles: these are typically interpreted using some variant of bar recursion, and appreciating the resulting realizer on a 'mathematical' level can be quite difficult.

The central theme that runs through this dissertation is to gain a better understanding of bar recursion and its role in extracting computational content from proofs in mathematical analysis. We organise our work into two parts. In the first part we present a series of case studies in which we use Gödel's functional interpretation extract and analyse bar recursive realizers from several well-known non-constructive principles. Our aim is to describe how these realizers work, and how they relate to the non-constructive ideas they interpret. The second part is more abstract, and focuses on the relationship between different variants of bar recursion. We establish a number of new definability results, and combining these with existing research manage to classify most of the well-known known modes of bar recursion used in proof theory into two main groups according to whether they are equivalent to Spector's bar recursion, or modified bar recursion.

1.1 The computational content of proofs in classical analysis

The computational meaning of non-constructive proofs is an issue that lies at the heart of mathematical logic and theoretical computer science, and one which has been continually studied in one form or another since the early twentieth century. By now there exists a plethora of elegant solutions to this problem along with corresponding methods for extracting computational content from classical proofs. These include traditional methods such as cut elimination and Hilbert's ε-calculus, more recent interpretations of classical logic in terms of game semantics [8, 22], or learning procedures [3, 4], and last but not least proof interpretations, which include Gödel's functional interpretation combined with the negative translation [37], and intuitionistic realizability combined with the A-translation [34, 54].

In this dissertation we focus largely on just one of these methods, Gödel's functional interpretation, and in the context of its computational interpretation of full classical analysis. A few years after the publication of Gödel's original paper on the functional interpretation of Peano arithmetic in the system T of primitive recursive functionals, Spector extended the functional interpretation to classical analysis by adding to system T constants for *bar recursion*, a new form of recursion over well-founded trees that solves the functional interpretation of the double-negation shift, and therefore allows the interpretation of both countable and countable dependent choice [86].

To date, relatively few concrete examples of program extraction involving Spector's bar recursion exist in the literature, and those that do (e.g. [49, 78]) are generally less concerned with the semantics of these realizers and more with their contribution to the complexity of bounds extracted for Π_2-theorems. However, by making an effort to understand the semantics of bar recursion we create the opportunity to gain genuine insight into the constructive meaning of the central principles of mathematical analysis.

Unfortunately, in practice, extracting meaningful computational content from non-trivial proofs can pose quite a challenge. We are required to formalise a proof to some extent before we can apply a proof interpretation, and in doing so we convert it from intuitive day-to-day mathematical language to the more abstruse syntax of formal logic. As a result, our extracted realizer often ends up as a complex higher-type term that reflects this formal proof as opposed to the mathematical one. Identifying the salient features of the realizer and 'decoding' it so that it can be read and understood as a constructive proof in the language of mathematics is a difficult task, especially so when our realizer involves non-trivial forms of recursion such as bar recursion.

The first part of this dissertation concentrates on making sense, in qualitative terms, of bar recursive realizers extracted from classical proofs, through a small collection of case studies. Our main tool is to use Spector's bar recursion in the form of the recently developed product of selection functions [29, 30] which also computes optimal strategies in a class of sequential games and allows us to impose onto our realizers an intuitive game semantics.

In our first and shortest case study (Chapter 6) we analyse the simple bar recursive realizer of arithmetic comprehension for Σ^0_1-formulas. As well as making explicit the game semantics of this realizer, we give a description of its *operational* behaviour and show that it carries out an intuitive 'learning process' in the course of constructing an approximation to the comprehension function.

We then move on to weak König's lemma for Σ^0_1-formulas (Chapter 7), and extract realizers for this principle that can be clearly described in game-theoretic terms. As an application we give intuitive computational interpretations to simple instances of the Bolzano-Weierstrass theorem and Ramsey's theorem for pairs, both of which have been recently analysed in the context of proof mining [55, 57, 78].

Finally, in our most intricate case study (Chapter 8) we extract a new realizer for the functional interpretation of the minimal-bad-sequence argument for arbitrary well-founded relations, using symbols $\mathsf{R}_<$ for transfinite recursion to interpret the least element principle. We show that our realizer can be clearly understood as a finitary analogue of the minimal-bad-sequence construction that implements backtracking and well-founded recursion to build an approximation to a minimal bad sequence. We then recall Berger's concise open recursive solution to the realizability interpretation of the classically equivalent principle of open induction, and discuss the possibility of devising a new form of open recursion that realizes the functional interpretation of both open induction and the minimal-bad-sequence argument. We conclude with an application in which we extract a new realizer for Higman's lemma.

1.2 Comparing variants of bar recursion

Following Spector's fundamental work on bar recursion, several alternative forms of bar recursion over well-founded trees have been developed, typically to give computational interpretations to analysis in new situations. These variants of bar recursion differ in interesting ways.

While Spector's bar recursion involves an explicit stopping condition $\varphi(\widehat{s}) < |s|$, variants of bar recursion given in [8] and [11] that witnesses realizability interpretations of countable dependent choice are *implicitly* well-founded by a continuity argument. More radical departures from bar recursion exist in the form of the so-called BBC functional [8] which involves out a symmetric 'demand driven' algorithm, or Berger's open recursion

[10] which carries out recursion over lexicographic orderings on sequences.

The theoretical part of the dissertation concerns the classification of modes of bar recursion according to whether or not they are interdefinable. We establish several new definability results.

Chapter 10 focuses on finite bar recursion and its relationship to Gödel's primitive recursive functionals. We define a hierarchy of fragments of system T based on finite bar recursion and show that they are in one-to-one correspondence with the usual fragments based on restricted primitive recursion.

In the original paper on bar recursion [86] Spector defines a general instance of bar recursion GBR but points out that only a restricted form SBR is necessary to interpret classical analysis. Chapter 11 contains a short proof that actually, SBR is strong enough to primitive recursively define GBR, and so the two are equivalent over system T.

Our main contribution, in Chapter 12, is to prove that open recursion, update recursion and the symmetric BBC functional are all primitive recursively equivalent to the implicitly iterated product of selection functions, which as shown in [30] is equivalent to modified bar recursion. These results are the first to relate the 'sequential' variants of bar recursion which include modified bar recursion and the product of selection functions to the 'open recursive' variants which include open recursion and the BBC functional.

1.3 Overview of the dissertation

We now give a concise map of the dissertation as a whole, and provide details of the collaborations involved in its authorship.

Part I consists of standard background material.

- **Chapter 2** contains an outline of the formal systems of arithmetic and analysis used in the dissertation and contains a list of our notational conventions.
- **Chapter 3** forms a short introduction to Gödel's functional interpretation, together with some concrete illustrations.
- **Chapter 4** contains several crucial results on Spector's bar recursion that will be needed in later chapters.

Part II contains our practical contributions on the extraction of realizers from proofs in classical analysis.

- **Chapter 5** reviews the product of selection functions as defined by Escardó and Oliva. We introduce a new sequential variant of dependent choice that will be required in Chapter 8 and show that it is naturally interpreted by the dependent product of selection functions.

- **Chapter 6** is concerned with the functional interpretation of Σ^0_1-arithmetical comprehension. We show that the usual bar recursive realizer of comprehension can be characterised in terms of a learning-process.
- **Chapter 7** focuses on the computational content of Σ^0_1-WKL. We extract game-theoretic realizers for the Bolzano-Weierstrass theorem and Ramsey's theorem as applications. This chapter was done jointly with Paulo Oliva and is published as [68, 69].
- **Chapter 8** examines the computational content of the minimal-bad-sequence argument. We extract a bar recursive realizer from its classical proof, and later consider an new, direct realiser in terms of open recursion. We give a new computational interpretation of Higman's lemma as an application. This chapter forms an extended version of [74].

Part III contains our theoretical contributions on the interdefinability of variants of bar recursion.

- **Chapter 9** provides a uniform presentation of the variants of bar recursion considered in subsequent chapters and summarises known definability results.
- **Chapter 10** focuses on finite bar recursion. We define a hierarchy of fragments of system T based on finite bar recursion and proves they are in one-to-one correspondence with the usual fragments based on primitive recursion. We also calibrate which fragments are needed to interpret bounded collection axioms. This chapter is based on joint work with Martín Escardó and Paulo Oliva published in [32].
- **Chapter 11** consists of a proof that Spector's general form of bar recursion GBR can be defined from the special form SBR. This is result is joint work with Paulo Oliva, and published in [70].
- **Chapter 12** contains a proof that the BBC functional, update recursion, open recursion and the product of selection functions IPS are primitive recursively equivalent in the type-structures of continuous functionals. This amounts to showing that IPS defines open recursion, and BBC defines IPS.

1.4 List of publications

The following publications contain some of the research reported in this dissertation.

[32] **System T and the product of selection functions**

Proceedings of Computer Science Logic (CSL'11), LIPIcs 12:233-247, 2011. (with Martín Escardó and Paulo Oliva)

[68] **A constructive interpretation of Ramsey's theorem via the product of selection functions**

To appear: Mathematical Structures in Computer Science, 2013. (with Paulo Oliva)

[69] **A game-theoretic interpretation of proofs in classical analysis**

To appear: Gentzen Centenary Volume, 2013. (with Paulo Oliva)

[70] **On Spector's bar recursion**

Mathematical Logic Quarterly, 58(4-5):356-365, 2012. (with Paulo Oliva)

[74] **Applying Gödel's Dialectica interpretation to obtain a constructive proof of Higman's lemma**

Proceedings of Classical Logic and Computation (CL&C'12), EPTCS 97:49-62, 2012.

I Background

CHAPTER

TWO

Formal Theories

We begin with a brief outline of the formal systems that are involved in this dissertation. Everything in this chapter is entirely standard, and the reader is directed to e.g. [18, 48, 90] for further details. Our main purpose here is to allow the reader to become familiar with the specific notation and conventions that will be used in subsequent chapters.

2.1 First order Heyting and Peano arithmetic

The language $\mathcal{L}(\mathsf{HA})$ of intuitionistic, or Heyting arithmetic HA contains the standard logical constants, along with number variables $x, y, z, \ldots$, a non-logical constant 0, a unary function symbol S and function symbols for all primitive recursive functions, along with a binary predicate $=$ for equality between numbers. The axioms and rules of HA consist of the usual axioms and rules of intuitionistic logic, axioms for S and $=$, defining axioms for the primitive recursive functions and finally the axiom schema of induction

$$\mathsf{IND}\ :\ A(0) \wedge \forall m(A(m) \to A(m+1)) \to \forall m A(m)$$

where A is any formula in $\mathcal{L}(\mathsf{HA})$. Classical, or Peano arithmetic PA is obtained from HA by adding the law of excluded middle for arbitrary formulas.

Fragments of Peano arithmetic

One can construct a variety of fragments of arithmetic based on restricted induction or restricted forms of other strong axioms. Alternatives to induction include the axiom schema of *finite choice*

$$\mathsf{FAC}\ :\ \forall i \leq m \exists x A(i, x) \to \exists s \forall i \leq m A(i, s_i),$$

where s is a finite sequence of numbers of length $m+1$ (suitably encoded as a single number), or the more widely used *bounded collection* axiom

$$\mathsf{BC}\ :\ \forall i \leq m \exists x A(i,x) \rightarrow \exists t \forall i \leq m \exists x \leq t A(i,x).$$

The proof theory of Peano arithmetic and its fragments forms a rich and extensive area of research (see [18] for an introduction). In this dissertation the fragments of arithmetic will only feature in Chapter 10, and we restrict ourselves to presenting only some basic facts here.

Definition 2.1. The class of bounded formulas Δ_0 consists of all formulas that are quantifier-free save for the bounded quantifiers $\forall i \leq t$ and $\exists i \leq t$. The classes of Π_n and Σ_n formulas are defined inductively for all $n \geq 0$ as follows:

(i) $\Pi_0 = \Sigma_0 = \Delta_0$;

(ii) Π_{n+1} consists of all formulas of the form $\forall \underline{x} A$ where A is a Σ_n-formula and $\underline{x}$ is a possibly empty tuple of variables;

(iii) Σ_{n+1} consists of all formulas of the form $\exists \underline{x} A$ where A is a Π_n-formula and $\underline{x}$ is a possibly empty tuple of variables.

Definition 2.2 (Fragments of arithmetic). The axiom schema Φ-IND denotes the axiom of induction restricted to Φ-formulas, where Φ is any of the classes defined in Definition 2.1. The axioms Φ-FAC and Φ-BC are defined similarly.

- A weak induction-free fragment of arithmetic PA_0 is usually defined to contain only some basic non-logical symbols 0, S, $+$, $\cdot$ and $\leq$ and their defining axioms, along with Δ_0-IND (there are various ways of formulating PA_0 precisely, among the most commonly used is Robinson's theory Q [89]).

- For $n \geq 1$ the theories $\mathsf{I}\Sigma_n$, $\mathsf{F}\Sigma_n$ and $\mathsf{B}\Sigma_n$ consist of PA_0 along with symbols and defining axioms for all primitive recursive functions, plus the axiom Σ_n-IND, Σ_n-FAC and Σ_n-BC respectively. The theories $\mathsf{I}\Pi_n$, $\mathsf{F}\Pi_n$ and $\mathsf{B}\Pi_n$ are defined analogously.

- Full Peano arithmetic PA consists of PA_0 plus all primitive recursive functions, along with induction for arbitrary formulas (or equivalently - by Theorem 2.3 - finite choice or bounded collection for arbitrary formulas).

Theorem 2.3. *Let $S \subseteq T$ mean that every theorem of T is a theorem of S, and write $S = T$ if the converse also holds. Then $\mathsf{I}\Sigma_n = \mathsf{I}\Pi_n$, $\mathsf{F}\Sigma_{n+1} = \mathsf{F}\Pi_n$ and $\mathsf{B}\Sigma_{n+1} = \mathsf{B}\Pi_n$. Moreover*

(a) $\mathsf{F}\Pi_n = \mathsf{B}\Pi_n$,

(b) $\mathsf{B}\Pi_n \subseteq \mathsf{I}\Sigma_{n+1}$ *but* $\mathsf{I}\Sigma_{n+1} \not\subseteq \mathsf{B}\Pi_n$*,*

(c) $\mathsf{I}\Sigma_n \subseteq \mathsf{B}\Pi_n$ *but* $\mathsf{B}\Pi_n \not\subseteq \mathsf{I}\Sigma_n$*.*

Proof. First proved in [71] and [72] (item (a) is due to [83]). See [18] for details. □

Theorem 2.3 shows that PA contains two strictly interleaving hierarchies, those based on induction and those based on finite choice/collection respectively.

$$\begin{tikzcd}
\mathsf{PA}_0 & \mathsf{I}\Sigma_n \arrow[l, dotted] \arrow[d, leftrightarrow] & & \mathsf{B}\Pi_n \arrow[ll] \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] & \mathsf{F}\Pi_n \arrow[d, leftrightarrow] & & \mathsf{I}\Sigma_{n+1} \arrow[ll] \arrow[d, leftrightarrow] & \mathsf{PA} \arrow[l, dotted] \\
& \mathsf{I}\Pi_n & & \mathsf{B}\Sigma_{n+1} \arrow[r, leftrightarrow] & \mathsf{F}\Sigma_{n+1} & & \mathsf{I}\Pi_{n+1} &
\end{tikzcd}$$

Figure 2.1: Fragments of Peano arithmetic (where $A \to B$ denotes $A \supseteq B$)

2.2 Extensional Heyting and Peano arithmetic in all finite types

In most of this dissertation we work in a standard higher-type extension of Peano arithmetic. This enables us to properly formalize proofs in mathematics, and to describe and reason about the higher-type forms of recursion needed to give these proofs a computational interpretation. The finite types are generated inductively as

$$\mathsf{types} := \mathbb{N} \mid \rho \to \tau \mid \rho \times \tau \mid \rho^*$$

where $\rho \to \tau$ (which we may also write as τ^ρ) is the type of functions mapping objects of type ρ to objects of type τ, $\rho \times \tau$ is the cartesian product of ρ and τ, and ρ^* the type of finite sequences whose elements have type ρ. We write x^ρ or $x\colon \rho$ to denote that an object x has type ρ. The degree $\deg(\rho)$ of the type ρ is defined inductively as

$$\begin{aligned}
\deg(\mathbb{N}) &:= 0, \\
\deg(\rho \to \tau) &:= \max\{\deg(\rho) + 1, \deg(\tau)\} \\
\deg(\rho \times \tau) &:= \max\{\deg(\rho), \deg(\tau)\}, \\
\deg(\rho^*) &:= \deg(\rho)
\end{aligned}$$

It will be useful to also consider a type $\mathbb{B}$ of booleans, and to identify two subsets of the finite types we which called the *compact* and *discrete* types, after [27]:

$$\begin{aligned}
\mathsf{compact} &:= \mathbb{B} \mid \mathsf{compact} \times \mathsf{compact} \mid \mathsf{discrete} \to \mathsf{compact} \\
\mathsf{discrete} &:= \mathbb{B} \mid \mathbb{N} \mid \mathsf{discrete} \times \mathsf{discrete} \mid \mathsf{discrete}^* \mid \mathsf{compact} \to \mathsf{discrete}.
\end{aligned}$$

The theory $\mathsf{E\text{-}HA}^\omega$ of extensional Heyting arithmetic in all finite types is based on that described in [48, 90]. The language of $\mathsf{E\text{-}HA}^\omega$ contains variables $x^\rho, y^\rho, z^\rho, \ldots$ and quantifiers $\forall/\exists x^\rho$ for each type ρ, along with the non-logical constants 0, $S^{\mathbb{N}\to\mathbb{N}}$ and the usual constructors including the projector $\Pi_{\rho,\tau}$ and combinator $\Sigma_{\sigma,\rho,\tau}$, basic operations for product and sequence types, the recursors R_ρ, and a binary predicate $=_\mathbb{N}$ for equality between numbers. Equality between objects of compound type is not taken as primitive but is defined inductively by

$$\begin{aligned} f =_{\rho\to\tau} g &:\equiv \forall x^\rho(fx =_\tau gx), \\ \langle x_0, x_1\rangle =_{\rho\times\tau} \langle y_0, y_1\rangle &:\equiv (x_0 =_\rho y_0 \wedge x_1 =_\tau y_1), \\ s =_{\rho^*} t &:\equiv (|s| = |t| \wedge \forall i < |s|(s_i =_\rho t_i)). \end{aligned}$$

The axioms and rules of $\mathsf{E\text{-}HA}^\omega$ contain the axioms and of rules of intuitionistic logic (in all finite types), defining axioms for each of the non-logical constants - including axioms for the recursor

$$\begin{aligned} \mathsf{R}_\rho^{y,z}(0) &=_\rho y \\ \mathsf{R}_\rho^{y,z}(n+1) &=_\rho z_n(\mathsf{R}_\rho^{y,z}(n)), \end{aligned}$$

axioms for $=_\mathbb{N}$, the axiom schema of induction

$$A(0) \wedge \forall m(A(m) \to A(m+1)) \to \forall m A(m).$$

for any formula A, and finally the axiom of extensionality:

$$\mathsf{EXT}_{\rho,\tau}\ : \forall t^{\rho\to\tau}, x^\rho, y^\rho(x =_\rho y \to tx =_\tau ty).$$

Extensional Peano arithmetic $\mathsf{E\text{-}PA}^\omega$ is obtained from $\mathsf{E\text{-}HA}^\omega$ by adding the law of excluded middle

$$A \vee \neg A$$

for arbitrary formulas.

Remark 2.4. We could have alternatively constructed $\mathsf{E\text{-}HA}^\omega$ over minimal collection types containing only $\mathbb{N}$ and function types $\rho \to \tau$, and encoded the types $\rho \times \tau$ and ρ^* as secondary constructions.

Remark 2.5. Tuples of variables $x^{\rho_0}, \ldots, x^{\rho_n}$ will always we regarded as a single variable $x\colon \rho_0 \times \ldots \times \rho_n$.

Remark 2.6. In the context of finite types $\Sigma_n^m(\Pi_n^m)$-formulas are $\Sigma_n(\Pi_n)$-formulas whose quantifiers are of type degree at most m.

Weakly extensional Heyting and Peano arithmetic

The weakly extensional variants $\mathsf{WE\text{-}HA}^\omega$ ($\mathsf{WE\text{-}PA}^\omega$) of $\mathsf{E\text{-}HA}^\omega$ ($\mathsf{E\text{-}PA}^\omega$) have the axiom of extensionality EXT replaced by the following rule of extensionality

$$\mathsf{QF\text{-}ER} : \ \frac{A_0 \to s =_\rho t}{A_0 \to r[s/x] =_\tau r[t/x]}$$

where A_0 is quantifier-free and s, t, r are arbitrary terms.

Features of $\mathsf{(W)E\text{-}HA}^\omega$

For a detailed account of the properties of $\mathsf{(W)E\text{-}HA}^\omega$ the reader is directed to [48, 90]. We recall that $\mathsf{WE\text{-}HA}^\omega$ admits standard constructions such as λ-abstraction (via the combinators $\Pi_{\rho,\tau}$, $\Sigma_{\sigma,\rho,\tau}$), definition by cases and bounded search. If $A_0(x^\rho)$ is a quantifier-free formula in the language of $\mathsf{WE\text{-}HA}^\omega$ with free variables x, then there exists a closed term $\chi\colon \rho \to \mathbb{B}$ such that

$$\mathsf{WE\text{-}HA}^\omega \vdash \forall x(\chi(x) =_{\mathbb{B}} 0 \leftrightarrow A_0(x)).$$

This means that quantifier-free formulas are decidable, and can be used - via their characteristic functions - as clauses in function definitions by cases. We permit quantifier-free formulas to contain *bounded* quantifiers $\forall/\exists x \leq t$, as these can be eliminated using bounded search.

Our theory $\mathsf{WE\text{-}HA}^\omega$ contains the following terms, formulas and abbreviations:

- For each type ρ constant functions n_ρ for $n \in \mathbb{N}$, defined inductively by $n_{\mathbb{N}} := n$ and $n_{\rho\to\tau} := \lambda x.n_\tau$ (analogously for $\rho \times \tau$, ρ^*). The term 0_ρ will be taken as a canonical element of type ρ;

- Projections $\pi_i\colon \rho_0 \times \rho_1 \to \rho_i$ and pairing functions $\langle\ ,\ \rangle\colon \rho_0 \to (\rho_1 \to \rho_0 \times \rho_1)$ for each pair types ρ_0, ρ_1. Given $t\colon \rho_0 \times \rho_1$ we write $t_i := \pi_i t$, and sometimes for sequences $\alpha\colon (\rho_0 \times \rho_1)^{\mathbb{N}}$ we write $\alpha_i := \lambda n.\pi_i\alpha(n)$;

- Functions $|\cdot|\colon \rho^* \to \mathbb{N}$ and $*\colon \rho^* \times \rho^* \to \rho^*$ for each ρ where $|s|$ is the length of s and $s * t$ the concatenation of s and t. We will also use $*$ to denote the function $\rho^* \times \rho^{\mathbb{N}} \to \rho^{\mathbb{N}}$ where $s * \alpha$ is the concatenation of s with the infinite sequence α.

- An overwrite function $@ : \rho^* \times \rho^{\mathbb{N}} \to \rho^{\mathbb{N}}$ defined by

$$(s\ @\ \alpha)(n) := \begin{cases} s_i & \text{if } n < |s| \\ \alpha(n) & \text{if } n \geq |s| \end{cases}$$

- A bounded search operator $\mu\colon \mathbb{B}^{\mathbb{N}} \times \mathbb{N} \to \mathbb{N}$ satisfying

$$\mu(\alpha, n) := \begin{cases} \text{least } i \leq n(\alpha i = 0) & \text{if it exists} \\ 0 & \text{otherwise.} \end{cases}$$

- We write $s \prec t$ or $s \preceq t$ if s is a (proper) initial segment of t. We write $s \prec \alpha$ if s is an initial segment of the infinite sequence α.

- For $\alpha\colon \rho^{\mathbb{N}}$ we define $[\alpha](n) := \langle \alpha(0), \ldots, \alpha(n-1) \rangle$ and for $s\colon \rho^*$ we define $\widehat{s} := s \mathbin{@} 0_{\rho^{\mathbb{N}}}$. We define $\overline{\alpha, n} := \widehat{[\alpha](n)}$.

- For a function $q\colon \rho^{\mathbb{N}} \to \mathbb{N}$, we define $q_s\colon \rho^{\mathbb{N}} \to \mathbb{N}$ by $q_s(\alpha) := q(s * \alpha)$.

- For $\alpha\colon \rho^{\mathbb{N}}$ we define $\mathsf{tail}_k \alpha := \lambda n.\alpha(k + n)$.

- The formula $n \in [x, y)$ for $n, x, y\colon \mathbb{N}$ is an abbreviation for $x \leq n < y$. We occasionally quantify over $[x, y)$ and write $\forall/\exists n \in [x, y)$, and analogously for (x, y), $(x, y]$, $[x, y]$.

- The term 'x^ρ if b' of type ρ is an abbreviation for $\begin{cases} x & \text{if } b = 0 \\ 0_\rho & \text{if } b \neq 0. \end{cases}$

Gödel's System T

The terms of $\mathsf{E\text{-}HA}^\omega$ are better known as the terms of Gödel's system T, and the set-theoretic functionals represented by the closed terms of $\mathsf{E\text{-}HA}^\omega$ the primitive recursive functionals in all finite types. Technically, the *theory* T as described in [37] corresponds to a quantifier-free fragment of $\mathsf{E\text{-}HA}^\omega$, but here we will always use $\mathsf{E\text{-}HA}^\omega$ to reason about terms of T. The class T_n is the subset of T in which recursion is restricted to types of degree $\leq n$, and in particular the closed terms of T_0 coincide with the ordinary primitive recursive functions. System T substantially expands T_0, the most well-known illustration of this being the Ackermann function, which does not exist in T_0 but can be easily defined in T_1.

Example 2.7 (Ackermann function). The Ackermann function $A\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ has defining equations

$$A(m, n) \overset{\mathbb{N}}{:=} \begin{cases} n + 1 & \text{if } m = 0 \\ A(m-1, 1) & \text{if } m > 0 \text{ and } n = 0 \\ A(m-1, A(m, n-1)) & \text{if } m > 0 \text{ and } n > 0 \end{cases}$$

It can be constructed using the recursor of type $\mathbb{N} \to \mathbb{N}$ as follows. Using $R_{\mathbb{N}}$ define

$$y(n) = n + 1$$
$$z(m, F)(n) = \begin{cases} F(1) & \text{if } n = 0 \\ F(z(m, F)(n-1)) & \text{if } n > 0 \end{cases}.$$

Then it is easy to check that $\lambda m, n.R_{\mathbb{N}\to\mathbb{N}}yzm(n)$ satisfies the defining equations of the Ackermann function.

2.3 Subsystems of analysis

We now discuss the subsystems of mathematics in all finite types which we will used in practise to formalise mathematical proofs. These will be obtained from Peano arithmetic in finite types by adding various choice principles.

A base system of analysis

We obtain a good base system of mathematical analysis by adding to $\mathsf{E\text{-}PA}^\omega$ or $\mathsf{WE\text{-}PA}^\omega$ the axiom schema of quantifier-free choice:

$$\mathsf{QF\text{-}AC}^{\rho,\tau} \ : \ \forall x^\rho \exists y^\tau A_0(x, y) \to \exists f^{\rho\to\tau} \forall x A_0(x, fx)$$

for A_0 quantifier-free. The theory $\mathsf{E\text{-}PA}^\omega + \mathsf{QF\text{-}AC}^{\mathbb{N},\mathbb{N}}$ already contains the well-known subsystem of second order arithmetic RCA_0 as defined in [84], and is in fact considerably stronger since RCA_0 only admits induction for Σ_1-formulas (on the other hand, the fragment of $\mathsf{E\text{-}PA}^\omega + \mathsf{QF\text{-}AC}^{\mathbb{N},\mathbb{N}}$ with induction restricted to Σ_1-formulas and recursion restricted to type 0 is a conservative extension of RCA_0).

Unfortunately, full extensionality becomes problematic if we want a theory of analysis that admits a computational interpretation (see [48, Chapter 8]). In this case, unless we are able to eliminate extensionality in some way, we are forced to work in the weakly extensional variants of our subsystems. In this dissertation we make an effort to formalise proofs in the weakly extensional systems, although later on in Part III we allow ourselves full extensionality to reason about modes of higher-type recursion.

A sizeable portion of mathematics can be formalised in $\mathsf{WE\text{-}PA}^\omega + \mathsf{QF\text{-}AC}$. Note that in particular we can construct the rational numbers $\mathbb{Q}$ and the real numbers $\mathbb{R}$ via suitable encodings over $\mathbb{N}$ and $\mathbb{N} \to \mathbb{N}$ respectively, define the usual operations ($=_{\mathbb{Q}}$,$\leq_{\mathbb{Q}}$, $=_{\mathbb{R}}$, $\leq_{\mathbb{R}}$ etc.) and prove basic properties about them. For full details of the formalization of $\mathbb{Q}$ and $\mathbb{R}$ within $\mathsf{WE\text{-}HA}^\omega$ see [48, Chapter 4]. Here we take for granted that our base theory allows us to use a fairly expressive mathematical language. We occasionally treat $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{Q}^{\mathbb{N}}$, $\mathbb{R}^{\mathbb{N}}$ etc. as types on the assumption that these 'types' can be encoded by a suitable member of the usual types.

Weak König's lemma

A binary tree T is a predicate on sequences of Booleans $\mathbb{B}^*$ that is prefix-closed:

$$\forall s^{\mathbb{B}^*}, t^{\mathbb{B}^*}(T(s * t) \to T(s)).$$

If $T(s)$ holds we say that s is a *branch* of T. Unless otherwise stated, we assume that T is a decidable predicate. A tree is *infinite* if it contains branches of arbitrary length. *Weak König's lemma* is the statement that every infinite binary tree has an infinite branch, given formally by the schema

$$\mathsf{WKL} \ : \ \forall n \exists s(|s| = n \wedge T(s)) \to \exists \alpha^{\mathbb{B}^{\mathbb{N}}} \forall n T([\alpha](n)),$$

where T ranges over all decidable binary trees.

The importance of Weak König's lemma lies in the fact that it allows us to construct an 'intermediate' subsystem of analysis. The theory $\mathsf{E\text{-}PA}^\omega + \mathsf{QF\text{-}AC}^{\mathbb{N},\mathbb{N}} + \mathsf{WKL}$ contains the subsystem WKL_0 central to the field of reverse mathematics, in which we are, for instance, able to reason about Heine/Borel compactness but *not* sequential compactness or completeness [84].

From the perspective of proof mining, WKL is significant because effective bounds can still be extracted from proofs using WKL that are *prima facie* non-constructive (see [48, 51]).

Full classical analysis

A strong theory of classical analysis can be achieved through the addition of the axiom schema of *countable dependent choice*

$$\mathsf{DC}^\rho \ : \ \forall n, x^\rho \exists y^\rho A_n(x, y) \to \forall x_0 \exists f^{\mathbb{N} \to \rho}(f(0) = x_0 \wedge \forall n A_n(f(n), f(n+1))).$$

Weaker than DC is the axiom schema of *countable choice*

$$\mathsf{AC}^{\mathbb{N},\rho} \ : \ \forall n \exists x^\rho A_n(x) \to \exists f^{\mathbb{N} \to \rho} \forall n A_n(f(n)).$$

and in turn an instance of $\mathsf{AC}^{\mathbb{N},\mathbb{N}}$ proves the schema of *full comprehension* over numbers

$$\mathsf{CA}^{\mathbb{N}} \ : \ \exists f^{\mathbb{N} \to \mathbb{N}} \forall x(f(x) \leftrightarrow A(x)).$$

Already $\mathsf{E\text{-}PA}^\omega + \mathsf{QF\text{-}AC}^{\mathbb{N},\mathbb{N} \to \mathbb{N}} + \mathsf{QF\text{-}AC}^{\mathbb{N} \to \mathbb{N},\mathbb{N}} + \mathsf{CA}^{\mathbb{N}}$ contains the whole of second order arithmetic in the sense of [84], and is therefore capable of formalising most of classical analysis. The special case of *arithmetical comprehension* over numbers is given by

$$\mathsf{CA}_{\mathrm{ar}} \ : \ \exists f^{\mathbb{N} \to \mathbb{N}} \forall x(f(x) \leftrightarrow A_{\mathrm{ar}}(x)).$$

where A_{ar} is an *arithmetical* formula (containing only quantifiers of type level 0). The theory $\mathsf{E\text{-}PA}^\omega + \mathsf{QF\text{-}AC}^{\mathbb{N},\mathbb{N}} + \mathsf{CA}_{\mathrm{ar}}$ contains the subsystem of second order arithmetic ACA_0.

CHAPTER

THREE

Proof Interpretations

In this chapter we provide some basic background material on proof interpretations, our main focus being on Gödel's functional (or 'Dialectica') interpretation. The reader is encouraged to consult [5, 48, 90] for details and a much more comprehensive treatment than that given here.

Gödel's functional interpretation belongs to a wider family of proof interpretations (also including Kreisel's modified realizability [54]) that consist of:

1. A mapping $I\colon \mathcal{T} \to \mathcal{F}$ from a theory $\mathcal{T}$ to a constructive system of higher-type functionals $\mathcal{F}$, under which formulas A in the language of $\mathcal{T}$ are mapped to formulas of the form $A^I :\equiv \exists x A_I(x)$ in the language of $\mathcal{F}$ satisfying
$$A \leftrightarrow \exists x A_I(x)$$
over some reasonable theory;

2. A soundness proof for I which contains an algorithm for converting a proof p of A in $\mathcal{T}$ to a *constructive* proof p^I of A^I in $\mathcal{F}$, by providing a direct interpretation to the axiom and rules of $\mathcal{T}$ and allowing us to unwind p^I by recursion over the structure of p. As a result, we obtain
$$\mathcal{T} \vdash A \Rightarrow \mathcal{F} \vdash A_I(t)$$
where t is a closed term of $\mathcal{F}$ that can be extracted from a given proof of A.

Gödel had devised his functional interpretation by 1941, although it was not published until his paper of 1958 [37], in which it is shown that Heyting and Peano arithmetic have a functional interpretation in the primitive recursive functionals of finite type, where in the case of Peano arithmetic one pre-composes the functional interpretation with the negative translation. A few years later Spector extended Gödel's result to full classical analysis by demonstrating that the negative translation of the axiom of choice has a Dialectica interpretation in the so-called bar recursive functionals of finite type [86]. These results can be easily adapted to incorporate the weakly-extensional higher-type subsystems of analysis based on $\mathsf{WE\text{-}PA}^\omega + \mathsf{QF\text{-}AC}$ ([90]).

classical theory	interpreting theory in all finite types
pure logic	λ-calculus (+ basic recursive functionals)
arithmetic	primitive recursive functionals
analysis	bar recursive functionals

Table 3.1: Functional interpretation of subsystems of mathematics

Proof interpretations like the functional interpretation were originally designed to obtain relative consistency proofs: If $\mathcal{T}$ is interpreted in $\mathcal{F}$ then the consistency of $\mathcal{T}$ follows from that of $\mathcal{F}$. However, since then they have found numerous applications in proof theory and in the process have been extended and refined in quite sophisticated ways. Notable applications include the establishment of conservation and closure results, the characterisation of provably recursive functions and - perhaps most significantly - the systematic unwinding computational content from non-constructive proofs.

The basic idea behind the latter of these applications is that the interpretation $\exists x A_I(x)$ represents a reformulation of A that admits a direct computational interpretation (where in particular the interpretation of Π_2-formulas $\forall x \exists y A_0(x,y)$ is given by $\exists f \forall x A_0(x, fx)$). The soundness proof gives us a means of extracting an explicit witness t for $\exists x$ from a proof p of A. The complexity of this witness will be restricted according to exactly what was used in the formalisation of p, and moreover because it has been extracted directly from the translated proof p^I its syntax and behaviour will mirror in some sense the structure of p. In other words, t carries computational information about A that reflects the fact that that it was proved by restricted means rather than that it is simply true. A witness t can often be extracted even when the proof p involves ineffective principles, and it is this use of proof interpretations to extract non-trivial constructive information subtly hidden in classical proofs that forms the basis of the proof mining program, and also forms one of the central themes of this dissertation.

In the present chapter we review Gödel's functional interpretation of Peano arithmetic in all finite types, while Spector's interpretation of classical analysis will be dealt with in Chapter 4. As well as defining the interpretation, we endeavour in Section 3.3 to provide some insight into how the interpretation can be understood in intuitive terms. Finally, we briefly discuss modified realizability and related interpretations in Section 3.4.

3.1 The negative translation

There are several ways of constructing negative translations from classical logic to intuitionistic logic. The first was given by Kolmogorov [52], although variants were discovered independently by Gödel and Gentzen. In this chapter we give the version of Kuroda [60]. The reader is directed to [33] for more on the relationship between these different negative translations. For proofs of the results in this section see e.g. [48, Chapter 10]. Note that $\neg A :\equiv A \to \bot$ as usual.

Definition 3.1 (Negative translation). Let A be a formula in a theory based on intuitionistic logic in all finite types. The negative translation of A is defined as $A^{\mathrm{N}} :\equiv \neg\neg A^*$ where A^* is defined recursively over the logical structure of A by

(i) $A^* :\equiv A$ if A is prime,

(ii) $(A \Box B)^* :\equiv A^* \Box B^*$ where $\Box \in \{\wedge, \vee, \to\}$,

(iii) $(\exists x^\rho A)^* :\equiv \exists x^\rho A^*$,

(iv) $(\forall x^\rho A)^* :\equiv \forall x^\rho \neg\neg A^*$.

Proposition 3.2. *For an arbitrary set of formulas* Δ *in the language of* $\mathsf{(W)E\text{-}PA}^\omega$ *we have*

$$\mathsf{(W)E\text{-}PA}^\omega + \Delta \vdash A \Rightarrow \mathsf{(W)E\text{-}HA}^\omega + \Delta^{\mathrm{N}} \vdash A^{\mathrm{N}}$$

where $\Delta^{\mathrm{N}} := \left\{B^{\mathrm{N}} \ : \ B \in \Delta\right\}$.

Whenever the negative translation in used in practise it is useful to simplify it if possible to avoid unnecessary double negations cluttering the formulas, and we assume familiarity with standard laws of intuitionistic logic in this respect.

Remark 3.3. In particular, note that intuitionistic logic proves

(i) $\neg\neg(A \to B) \leftrightarrow (A \to \neg\neg B) \leftrightarrow (\neg\neg A \to \neg\neg B)$,

(ii) $\neg\neg\forall x \neg\neg A(x) \leftrightarrow \forall x \neg\neg A(x)$,

while $\mathsf{WE\text{-}HA}^\omega$ proves

(iii) $\neg\neg A_0 \leftrightarrow A_0$ for A_0 quantifier-free.

The soundness of the negative translation can not be directly extended to our base system $\mathsf{WE\text{-}PA}^\omega + \mathsf{QF\text{-}AC}$ because $\mathsf{QF\text{-}AC}^{\mathrm{N}}$ is not provable in $\mathsf{WE\text{-}HA}^\omega + \mathsf{QF\text{-}AC}$. However, $\mathsf{QF\text{-}AC}^{\mathrm{N}}$ *can* be derived in the semi-classical theory $\mathsf{WE\text{-}HA}^\omega + \mathsf{QF\text{-}AC} + \mathsf{MP}^\omega$ where MP^ω is the Markov principle

$$\mathsf{MP}^\omega \ : \ \neg\neg\exists x^\rho A_0(x) \to \exists x A_0(x)$$

for $A_0(x)$ quantifier-free.

Proposition 3.4. *For an arbitrary set of formulas Δ in the language of* $\mathsf{WE\text{-}HA}^\omega$ *we have*

$$\mathsf{WE\text{-}PA}^\omega + \mathsf{QF\text{-}AC} + \Delta \vdash A \Rightarrow \mathsf{WE\text{-}HA}^\omega + \mathsf{QF\text{-}AC} + \mathsf{MP}^\omega + \Delta^N \vdash A^{\mathrm{N}}$$

where $\Delta^N := \{B^{\mathrm{N}} : B \in \Delta\}$. For the special case where Δ contains only purely universal formulas $\forall x^\rho B_0(x)$ note that Δ^{N} is equivalent to Δ over $\mathsf{WE\text{-}HA}^\omega$.

3.2 The functional interpretation

Gödel's functional interpretation, also widely known as the Dialectica interpretation, maps formulas A to formulas of the form $\exists x \forall y |A|^x_y$ where x, y are (possibly empty) sequences of variables and $|A|^x_y$ is a quantifier-free formula whose free variables are x, y and the free variables of A. We label x and y the *witness* and *counterexample* variables respectively. The idea is that A is equivalent to $\exists x \forall y |A|^x_y$, so that the functional interpretation carries out a kind of Skolemisation. We now give the precise definition.

Definition 3.5 (Functional 'Dialectica' interpretation). To every formula A in the language of $\mathsf{WE\text{-}HA}^\omega$ we assign a formula $A^D :\equiv \exists x \forall y |A|^x_y$, where $|A|^x_y$ is a quantifier-free formula defined by induction over the logical structure of A as follows.

(i) $|A| :\equiv A$ if A is prime,

(ii) $|A \wedge B|^{x,u}_{y,v} :\equiv |A|^x_y \wedge |B|^u_v$,

(iii) $|A \vee B|^{b,x,u}_{y,v} :\equiv (b = 0 \to |A|^x_y) \wedge (b \neq 0 \to |B|^u_v)$,

(iv) $|A \to B|^{U,Y}_{x,v} :\equiv |A|^x_{Yxv} \to |B|^{Ux}_v$,

(v) $|\exists t^\rho A(t)|^{z,x}_y :\equiv |A(z)|^x_y$,

(vi) $|\forall t^\rho A(t)|^X_{z,y} :\equiv |A(z)|^{Xz}_y$.

The definition of the functional interpretation is straightforward apart from the treatment of implication, which is essentially its characterising feature (see e.g. [67]). We travel from $\exists x \forall y |A|^x_y \to \exists u \forall v |B|^u_v$ to $\exists U, Y \forall x, v |A \to B|^{U,Y}_{x,v}$ via the following equivalences:

$$\begin{aligned}
& \exists x \forall y |A|^x_y \to \exists u \forall v |B|^u_v && (3.1)\\
\leftrightarrow\ & \forall x (\forall y |A|^x_y \to \exists u \forall v |B|^u_v) && (3.2)\\
\leftrightarrow\ & \forall x \exists u (\forall y |A|^x_y \to \forall v |B|^u_v) && (3.3)\\
\leftrightarrow\ & \forall x \exists u \forall v (\forall y |A|^x_y \to |B|^u_v) && (3.4)\\
\leftrightarrow\ & \forall x \exists u \forall v \exists y (|A|^x_y \to |B|^u_v) && (3.5)\\
\leftrightarrow\ & \exists U, Y \forall x \forall v (|A|^x_{Yxv} \to |B|^{Ux}_v). && (3.6)
\end{aligned}$$

While this route is non-constructive - in particular the passage from (3.2) to (3.3) requires independence of premise $\mathsf{IP}_\forall^\omega$ for universal formulas:

$$\mathsf{IP}_\forall^\omega \ : \ (\forall y C_0(y) \to \exists u D(u)) \to \exists u(\forall y C_0(y) \to D(u)),$$

(for C_0 quantifier-free) while (3.4) to (3.5) requires Markov's principle MP^ω - it is nevertheless the 'least non-constructive' Skolemisation of implication, which is precisely why it was chosen by Gödel.

One way to visualise $\exists x \forall y |A|_y^x$ is as a game played between the quantifiers ∃loise and ∀belard. The soundness theorem guarantees that whenever A is provable, ∃loise has a winning strategy t satisfying $\forall y |A|_y^t$. In this sense, game representing the functional interpretation of implication can be described as follows. First ∀belard plays a witness x for the premise of $|A|_y^x \to |B|_v^u$, challenging ∃loise to witness the conclusion. Once ∃loise has responded with a witness u, ∀belard then attempts to invalidate this witness with a counterexample v. Finally, ∃loise responds by trying to invalidate ∀belard's original witness with a counterexample y. The functional interpretation of implication is the statement that ∃loise has a winning strategy U, Y in this game.

Remark 3.6. Sometimes for notational convenience we omit the final step in the functional interpretation and give A^D as a Π_2 formula $\forall y \exists x A_0(y, x)$ when technically A^D is given by $\exists X \forall y A_0(y, Xy)$.

Interpreting intuitionistic arithmetic

Gödel's fundamental result of 1958 is that Heyting arithmetic has a functional interpretation in the system T primitive recursive functionals of finite type. His soundness proof remains valid we extend Heyting arithmetic with any new axioms which can be also be interpreted in system T. Further details and full proofs for this section can be found in e.g. [48, Chapters 8, 10].

Theorem 3.7 (Soundness theorem for Heyting arithmetic [37, 90])**.** *Let Δ be an arbitrary set of purely universal sentences in $\mathcal{L}(\mathsf{WE\text{-}PA}^\omega)$, and $A(a)$ be a formula in $\mathcal{L}(\mathsf{WE\text{-}PA}^\omega)$ containing only a free. Then*

$$\mathsf{WE\text{-}HA}^\omega + \mathsf{AC} + \mathsf{IP}_\forall^\omega + \mathsf{MP}^\omega + \Delta \vdash A(a) \Rightarrow \mathsf{WE\text{-}HA}^\omega + \Delta \vdash \forall y |A(a)|_y^{t(a)}$$

where t is a closed term of $\mathsf{WE\text{-}HA}^\omega$ which can be extracted from the proof of A.

Remark 3.8. Here, AC denotes the *full* axiom of choice in all types

$$\forall x^\rho \exists y^\tau A(x, y) \to \exists f^{\rho \to \tau} \forall x^\rho A(x, fx)$$

so in particular includes $\mathsf{QF\text{-}AC}$.

Remark 3.9. The Dialectica interpretation is *not* sound for fully extensional arithmetic $\mathsf{E\text{-}HA}^\omega$, which is a consequence of the fact that it interprets MP^ω (see [48, p.126]).

Outline of the proof of Theorem 3.7. Induction over the derivation of A. We illustrate only a few important cases. Firstly we note that only a typed λ-calculus (plus a few basic recursive functionals) is required to interpret the logical axioms and rules of $\mathsf{WE\text{-}HA}^\omega$. The most interesting case is the contraction axiom $A \to A \wedge A$. If A is interpreted as $|A|^x_y$ we must find terms t, t' and s satisfying

$$\forall x, v, v'(|A|^x_{sxvv'} \to |A|^{tx}_v \wedge |A|^{t'x}_{v'}).$$

It is clear that $t = t' := \lambda x.x$ and

$$sxvv' := \begin{cases} v & \text{if } \neg|A|^x_v \\ v' & \text{otherwise} \end{cases}.$$

do the trick. Note that we rely on the decidability of the quantifier-free formula $|A|^x_v$. This makes extending the functional interpretation to systems where decidability of quantifier-free formulas cannot be taken for granted quite problematic (see e.g. Burr [17] for an extension of the functional interpretation to set theory).

Primitive recursion is required to interpret the induction. To start with we can replace the induction axiom IND with the equivalent induction rule

$$\frac{A(0) \wedge \forall n(A(n) \to A(n+1))}{\forall n A(n)}\ \mathsf{IR}$$

which makes verifying the interpretation somewhat easier. Now, suppose we have interpreted the premise of the rule, namely we have t, Z and S satisfying

$$\begin{aligned} &\forall y |A(0)|^t_y \\ &\forall n, x, y(|A(n)|^x_{S_n xy} \to |A(n+1)|^{Z_n x}_y). \end{aligned}$$

Then we define $T0 := t$ and $T(n+1) := Z_n(Tn)$ via primitive recursion, and T satisfies

$$\begin{aligned} &\forall y |A(0)|^{T0}_y \\ &\forall y |A(n)|^{Tn}_y \to \forall y |A(n+1)|^{T(n+1)}_y. \end{aligned}$$

By induction we see that $\forall n, y |A(n)|^{Tn}_y$ holds in $\mathsf{WE\text{-}HA}^\omega$, which is the functional interpretation of the conclusion.

The remaining non-logical axioms and rules, including AC, $\mathsf{IP}^\omega_\forall$ and MP^ω, and any additional universal sentences Δ, are trivially interpreted. □

Remark 3.10. Gödel's original soundess proof was carried out in the quantifier-free system T as opposed to $\mathsf{WE\text{-}HA}^\omega$, his aim being to reduce the consistency of Heyting arithmetic to the weakest system possible.

Interpreting classical arithmetic

The functional interpretation can be extended to classical arithmetic by precomposing the functional interpretation proper with the negative translation. For example, by Theorem 3.7 combined with Proposition 3.2 we obtain

$$\mathsf{WE\text{-}PA}^\omega \vdash A \Rightarrow \mathsf{WE\text{-}HA}^\omega \vdash A^{\mathrm{N}} \Rightarrow \mathsf{WE\text{-}HA}^\omega \vdash \forall y |A^{\mathrm{N}}|^t_y.$$

We refer to the two-step interpretation as the ND-interpretation, or sometimes just the functional interpretation when it is clear that we are referring to its combination with the negative translation.

Theorem 3.11 (Soundness theorem for classical arithmetic)**.** *Let Δ be an arbitrary set of purely universal sentences in $\mathcal{L}(\mathsf{WE\text{-}PA}^\omega)$, and $A(a)$ be a formula in $\mathcal{L}(\mathsf{WE\text{-}PA}^\omega)$ containing only a free. Then*

$$\mathsf{WE\text{-}PA}^\omega + \mathsf{QF\text{-}AC} + \Delta \vdash A(a) \Rightarrow \mathsf{WE\text{-}HA}^\omega + \Delta \vdash \forall y |A(a)^{\mathrm{N}}|^{t(a)}_y$$

where t is a closed term of $\mathsf{WE\text{-}HA}^\omega$ which can be extracted from the proof of A.

Proof. By Proposition 3.4

$$\mathsf{WE\text{-}PA}^\omega + \mathsf{QF\text{-}AC} + \Delta \vdash A(a) \Rightarrow \mathsf{WE\text{-}HA}^\omega + \mathsf{QF\text{-}AC} + \mathsf{MP}^\omega + \Delta \vdash A(a)^{\mathrm{N}},$$

and therefore the result follows from Theorem 3.7 applied to $A(a)^{\mathrm{N}}$. □

The following fundamental result confirms that the ND-interpretation can directly extract programs from proofs of Π_2-theorems, even when those proofs are non-constructive.

Theorem 3.12 (Program extraction theorem)**.** *Let Δ be a set of purely universal sentences in $\mathcal{L}(\mathsf{WE\text{-}PA}^\omega)$, and $A_0(x^\rho, y^\tau)$ a quantifier-free formula in $\mathcal{L}(\mathsf{WE\text{-}PA}^\omega)$ with free variables x, y of arbitrary type. Then*

$$\mathsf{WE\text{-}PA}^\omega + \mathsf{QF\text{-}AC} + \Delta \vdash \forall x^\rho \exists y^\tau A_0(x, y) \Rightarrow \mathsf{WE\text{-}HA}^\omega + \Delta \vdash \forall x A_0(x, fx)$$

where f is a closed term of $\mathsf{WE\text{-}HA}^\omega$ which can be extracted from the proof of $\forall x \exists y A_0(x, y)$.

Proof. This follows from the crucial fact that the functional interpretation admits Markov's principle:

$$\begin{aligned}
&\mathsf{WE\text{-}PA}^\omega + \mathsf{QF\text{-}AC} + \Delta \vdash \forall x \exists y A_0(x, y)\\
&\overset{\text{Prop.3.4}}{\Rightarrow} \mathsf{WE\text{-}HA}^\omega + \mathsf{QF\text{-}AC} + \mathsf{MP}^\omega + \Delta \vdash \forall x \neg\neg\exists y A_0(x, y)\\
&\overset{\mathsf{MP}^\omega}{\Rightarrow} \mathsf{WE\text{-}HA}^\omega + \mathsf{QF\text{-}AC} + \mathsf{MP}^\omega + \Delta \vdash \forall x \exists y A_0(x, y)\\
&\overset{\text{Thm.3.7}}{\Rightarrow} \mathsf{WE\text{-}HA}^\omega + \Delta \vdash \forall x A_0(x, fx).
\end{aligned}$$

□

3.3 Understanding the Dialectica interpretation

The functional interpretation of classical logic translates a formula A into a variant $A^{\mathrm{ND}} \equiv \exists x \forall y |A^{\mathrm{N}}|^x_y$ that is weak enough to admit a direct computational interpretation, but still *classically* equivalent to A.

Theorem 3.13 (Characterisation theorem, Kreisel [54])**.** *We have*

$$\mathsf{WE\text{-}PA}^\omega + \mathsf{QF\text{-}AC} \vdash A \leftrightarrow \exists x \forall y |A^{\mathrm{N}}|^x_y$$

for any formula A in the language of $\mathsf{WE\text{-}PA}^\omega$.

While Theorem 3.13 verifies that A and A^{ND} are provably equivalent, a more intuitive understanding of this relationship can be quite difficult to attain. The combination of negative translation and functional interpretation is a subtle interpretation that maps even relatively simple theorems to abstruse higher-type functionals. The action of the ND interpretation on Π_2-formulas is straightforward enough to appreciate, but on the other hand Π_3-formulas of the form $A \equiv \forall u^\sigma \exists v^\rho \forall w^\tau A_0(u,v,w)$ are interpreted as

$$\exists V^{\sigma \times (\rho \to \tau) \to \rho} \forall u^\sigma, p^{\rho \to \tau} A_0(u, Vup, p(Vup)).$$

which even for arithmetical formulas already contains a functional V of degree 2. So how can we characterise the functional interpretation of A? We consider this question in three different contexts.

(a) *Proof theoretic.* Firstly we can try to appreciate why A and A^{ND} are logically equivalent. Suppose, for contradiction, that $\neg A$ holds. Then by $\mathsf{QF\text{-}AC}$ this is equivalent to the statement

$$\exists u, p^{\rho \to \tau} \forall v \neg A_0(u, v, pv),$$

where we envisage p as a ‘counterexample function’ that refutes any candidate realiser for $\exists v \forall w A_0(u,v,w)$. Therefore A is equivalent to

$$\neg \exists u, p^{\rho \to \tau} \forall v \neg A_0(u, v, pv)$$

which in turn is equivalent to

$$\forall u, p \exists v A_0(u, v, pv).$$

The ND interpretation of A asks for a functional V that witnesses this statement. For each parameter u, the role of Vu is to refute arbitrary counterexample functions p for $\exists v \forall w A_0(u,v,w)$. This rather intuitive ‘no-counterexample’ interpretation (n.c.i.) was studied by Kreisel in [53], where *arbitrary* formulas of PA are interpreted in this way by functionals of at most degree 2. The n.c.i. coincides with the functional interpretation for Π_3-formulas, but the two diverge for formulas of higher complexity. Unfortunately the restriction on types for the n.c.i. has the disadvantage in that its interpretation of implication is too weak to achieve the nice modular behaviour enjoyed by the functional interpretation (see [47]), although the intuition behind the n.c.i. is still beneficial when discussing the functional interpretation of relatively simple formulas.

(b) *Computational.* Suppose that A is an ineffective lemma used in the proof of a Π^0_2 statement i.e.

$$(*)\quad \forall u \exists v \forall w A_0(u,v,w) \to B :\equiv \forall x^{\mathbb{N}} \exists y^{\mathbb{N}} B_0(x,y).$$

The (partial) functional interpretation of $(*)$ given by

$$\forall V \exists f \forall x \exists u, p (A_0(u, Vup, p(Vup)) \to B_0(x, fx)),$$

describes a way of *using* the functional interpretation of the ineffective lemma A in the construction of a realizer $f_V \colon \mathbb{N} \to \mathbb{N}$ for B. The idea is that while A is needed to derive B, only an 'approximation' of A is required to prove $\exists y B_0(x,y)$ for *fixed* x. The functional V produces approximations to A whose 'quality' is determined by the parameters u, p, and the full functional interpretation of $(*)$ consists of the realizer f_V along with realizers $u_{V,x}$, $p_{V,x}$ that for each x calibrate the size of the approximation to A necessary to validate $B_0(x, f_V x)$. See also [48, p.171] for a more detailed discussion of this particular instance of the ND interpretation of implication.

(c) *Mathematical.* The reading of A^{ND} as a finitary approximation to A has been shown to be particularly illuminating in some mathematical contexts, in particular it has been observed in [48] that the functional interpretation is closely related to so called *correspondence principle* as discussed in [88]. This correspondence principle refers to the process of finitizing a 'soft' statement in mathematical analysis involving qualitative notions such as *infinite* or *convergent* to produce an equivalent 'hard' statement involving quantitative notions such as *large* or *metastable.* Roughly speaking, the former represent formulas in which the relationship between the quantifiers is hidden, the latter formulas in which this relationship is made explicit. In logical terms the process of finitization is essentially a shuffling of quantifiers so that they become dependent on one another - which is precisely what the ND interpretation does. The ND interpretation of certain theorems has been shown to coincide with the finitary versions of those theorems used in mathematics, albeit the former phrased in terms of higher-type functionals and the latter in terms of well-behaved set-theoretic functions. Generally, soft statements involve quantification over some compact space K which becomes eliminated during the finitisation process. From a proof theoretic perspective this corresponds to the ability of the *monotone* functional interpretation extract bounds for realizers that are uniform with respect to K. The relationship between the monotone interpretation and the correspondence principle is discussed in more detail in [36, 48].

We now illustrate our discussion of the functional interpretation with a well-known example that has been widely studied in the context of applied proof theory.

Example 3.14 (Cauchy convergence). Suppose we have a classical proof that a sequence (x_i) is Cauchy convergent, which can be formally stated as the Π^0_3 formula

$$\forall k \exists n \forall i, j \geq n (\|x_i - x_j\| \leq 2^{-k}), \tag{3.7}$$

or alternatively, absorbing the quantifiers $\forall i$, $\forall j$ into one, as

$$\forall k \exists n \forall l \forall i, j \in [n, n+l](\|x_i - x_j\| \leq 2^{-k}). \tag{3.8}$$

The ND interpretation of (3.8) is

$$\exists N^{\mathbb{N}\times(\mathbb{N}\to\mathbb{N})\to\mathbb{N}} \forall k, p^{\mathbb{N}\to\mathbb{N}} \forall i, j \in [N_{k,p}, N_{k,p} + p(N_{k,p})](\|x_i - x_j\| \leq 2^{-k}). \tag{3.9}$$

Following our previous discussion, one can view N_k as a functional whose job is to refute potential counterexample functions p which seek to disprove permanent stability within the error 2^{-k} by exhibiting, for each n, some $i, j \geq n$ with $\|x_i - x_j\| > 2^{-k}$.

From a slightly different perspective N_k is a functional that serves as an approximation to permanent stability of (x_i), for each k producing regions $[N_{k,p}, N_{k,p} + p(N_{k,p})]$ of *metastability* within error 2^{-k}. The statement (3.9) that such regions of metastability exist for any specified error 2^{-k} and for arbitrary p is fully equivalent to (3.8) and constitutes a finitary reformulation of the Cauchy convergence property. It turns out that this interpretation of Cauchy convergence plays a key role in ergodic theory, whereby one obtains quantitative versions of convergence theorems by extracting explicit bounds on N that are in some sense uniform with respect to sequences (x_i).

A simple example of this phenomenon is the monotone convergence principle, which states that any increasing sequence of rationals $0 \leq x_0 \leq x_1 \leq \ldots \leq 1$ is Cauchy convergent. This is easily formalised using classical logic: suppose for contradiction that there exists some k such that

$$\forall n \exists m (x_{n+m} - x_n > 2^{-k}).$$

Then by $\mathsf{QF\text{-}AC}^{\mathbb{N},\mathbb{N}}$ there exists a function p satisfying

$$\forall n (x_{n+p(n)} - x_n > 2^{-k}).$$

Defining $\tilde{p}(n) := n + p(n)$ we see that we must have $x_{\tilde{p}^{(i+1)}(0)} - x_{\tilde{p}^{(i)}(0)} \leq 2^{-k}$ for some $i \leq 2^k$ else $x_{\tilde{p}^{(2^k)}(0)} > 1$, and this contradicts the construction of p.

It has been shown that in general there is no computable bound on the rate of convergence of (x_i) [85]. However, the ND interpretation (3.9) of the monotone convergence principle is equivalent to

$$\exists N \forall k, p (x_{N_{k,p}+p(N_{k,p})} - x_{N_{k,p}} \leq 2^{-k}), \tag{3.10}$$

and from a quick inspection of our classical proof we obtain the computable bound $N_{k,p} \leq \tilde{p}^{2^k}(0)$. Furthermore, our bound on $N_{k,p}$ is independent of the sequence (x_i), so we can eliminate the implicit quantification over infinite increasing sequences and replace it with quantification over finite sequences of length $M_{k,p} := \tilde{p}^{2^k}(0)$, yielding

$$\forall k, p \forall (0 \leq x_0 \leq \ldots \leq x_{M_{k,p}} \leq 1) \exists n \leq n + p(n) \leq M_{k,p} \forall i, j \in [n, n+p(n)](|x_i - x_j| \leq 2^{-k}).$$

This is essentially an explicit version of the so called finite convergence principle considered by Tao in [88]:

> If k is a natural number, $p \colon \mathbb{N} \to \mathbb{N}$ a function and $0 \leq x_0 \leq \ldots \leq x_M \leq 1$ a finite sequence such that M is sufficiently large depending on k and p, then there exists n with $0 \leq n \leq n + p(n) \leq M$ such that $|x_i - x_j| \leq 2^{-k}$ for all $n \leq i, j \leq p(n)$.

The finite convergence principle is an important principle in ergodic theory that lies behind, for instance, the proof of the Szemeredi regularity lemma, and therefore in this case the functional interpretation yields a finitary reformulation of a non-constructive theorem that has genuine mathematical significance.

The extraction of uniform bounds for the functional interpretation of more complex convergence theorems has been the subject of several papers in the intersection of proof theory and ergodic theory. For instance, in [6] the functional interpretation is used to develop a quantitative version of the mean ergodic theorem.

Remark 3.15 (Infinite pigeonhole principle). Let $[k]$ denote the set $\{0, 1, \ldots, k-1\} \subset \mathbb{N}$. A finite colouring of the natural numbers with k colours is a function $c \colon \mathbb{N} \to [k]$. The infinite pigeonhole principle (IPHP) states that any finite colouring uses at least one colour infinitely often, or more formally

$$\mathsf{IPHP}[k, c] \ : \ \exists i \leq k \forall n \exists m (m \geq n \wedge c(m) = i). \tag{3.11}$$

One can show that there exist realizers for the functional interpretation of IPHP that are uniform with respect to the compact space $\mathbb{N} \to [k]$, and therefore similarly to the previous example the functional interpretation can be used to produce a finitary form of IPHP which one only needs to quantify over finite colourings. This finitization via the functional interpretation is discussed in [48, Chapter 2]. Interestingly, a finitization of IPHP 'by hand' without resorting to proof theoretic tools was also given by Tao in an early version of [88]. This latter finitization turned out to be false, and was later corrected in light of the functional interpretation. An account of the whole story and the precise relationship between each of these finitizations in given in [36], which more generally forms an interesting case study of the interplay between mathematics and proof theory.

3.4 Modified realizability, and other proof interpretations

While our main focus in this dissertation will be on the functional interpretation, many of the issues we discuss pertain to proof interpretations in general. We conclude the chapter with a brief mention of some other functional interpretations that feature in proof theory. Because Chapter 8 involves a comparison between the functional interpretation and Kreisel's modified realizability [54], which has also been widely used in program extraction, we take this opportunity to recall its definition.

Definition 3.16 (Modified realizability). To every formula A in the language of $\mathsf{E\text{-}HA}^\omega$ we assign a formula $A^{mr} :\equiv \exists x(x\ mr\ A)$, where $x\ mr\ A$ is a $\exists$-free formula defined by induction over the logical structure of A as follows.

(i) $x\ mr\ A :\equiv A$ if A is prime,

(ii) $x, y\ mr\ (A \wedge B) :\equiv x\ mr\ A \wedge y\ mr\ B$,

(iii) $b, x, y\ mr\ (A \vee B) :\equiv (b = 0 \to x\ mr\ A) \wedge (b \neq 0 \to y\ mr\ B)$,

(iv) $f\ mr\ (A \to B) :\equiv \forall x(x\ mr\ A \to fx\ mr\ B)$,

(v) $x\ mr\ (\forall t A(t)) :\equiv \forall t(xt\ mr\ A(t))$,

(vi) $z, x\ mr\ (\exists t A(t)) :\equiv x\ mr\ A(z)$.

Modified realizability differs from the functional interpretation most significantly in its treatment of implication - which is closer to the natural BHK interpretation of implication and somewhat easier to appreciate.

However, while $\mathsf{E\text{-}HA}^\omega$ has a straightforward realizability interpretation in the primitive recursive functionals analogous to Theorem 3.7 (see [48, Chapter 5]), in contrast to the functional interpretation modified realizability does not assign any constructive content to negated formulas. As a result, if we want to extend modified realizability to classical logic it is not enough to simply compose it with the negative translation - the typical solution to this has been to add Friedman's A-translation as an intermediate step [34].

This combination of modified realizability and the A translation has several drawbacks compared to the ND interpretation. In particular it loses modularity, and it is not even sound for the weak quantifier-free rule of extensionality QF-ER. Nevertheless, many refinements and improvements of modified realizability have been proposed, and it is still open whether or not there are situations in which modified realizability is better suited to the functional interpretation in program extraction.

In addition to the functional interpretation and modified realizability, notable proof interpretation include:

1. *The Diller-Nahm interpretation.* This hybrid of modified realizability and the functional interpretation adjusts the functional interpretation's treatment of implication to

   $$(A \to B)^{DN} :\equiv \exists U, Y \forall x, v (\forall y < |Yxv| (|A|^x_y \to |B|^{Ux}_v))$$

   where Y now returns a finite sequence as opposed to a single element. This interpretation can be viewed as a 'Dialectica-like' interpretation, which nevertheless shares nice structural properties with modified realizability: in particular, we no longer require decidability of formulas to interpret contraction. The Diller-Nahm interpretation plays an important role in the categorical semantics of proof interpretations given by the *Dialectica categories* (see [24]).

2. *The bounded and monotone variants of interpretations.* These interpretations ask for *bounds* on witnesses as opposed to exact witnesses. For instance, while the Dialectica interpretation extracts a term t satisfying $\forall y |A^{\mathrm{N}}|^t_y$, its monotone variant asks only for a bound t^* (or at higher types a *majorant*) satisfying

   $$\exists x (t^* \geq x \land \forall y |A^{\mathrm{N}}|^x_y).$$

   These interpretations are central to the proof mining program because by relaxing their requirement for for a precise witness they are capable of interpreting a wider range of non-constructive principles. For instance, WKL is trivially realized by the monotone functional interpretation, and therefore contributes nothing to the complexity of extracted bounds.

A uniform syntactic framework encompassing many of these interpretations has been developed by Oliva in [67].

CHAPTER

FOUR

Spector's Bar Recursion

The main result reviewed in the previous chapter was that classical arithmetic has an ND interpretation in the primitive recursive functionals of finite type. However, primitive recursion no longer suffices if we wish to expand the soundness theorem to incorporate strong comprehension or choice principles. In a remarkable paper of 1962 [86], C. Spector succeeded in extending Gödel's functional interpretation to full classical analysis by adding to system T constants for *bar recursion* in all finite types.

The bar recursive functional interpretation of classical analysis will be treated in detail in Chapter 5 using the product of selection functions. In this introductory chapter we simply prepare the way for later work by stating Spector's result and proving some important properties about bar recursion. We also mention Howard's realizer of weak König's lemma using a restricted, binary form of GBR. Finally, we take the opportunity to briefly introduce some well-known type structures of $\mathsf{E\text{-}HA}^{\omega} + \mathsf{GBR}$.

Spector's so called 'general' bar recursor $\mathsf{GBR}_{\rho,\tau}$ where ρ, τ are arbitrary types has defining axiom

$$\mathsf{GBR}^{\phi,q,\varphi}_{\rho,\tau}(s^{\rho^*}) \overset{\tau}{:=} \begin{cases} q(s) & \text{if } \varphi(\hat{s}) < |s| \\ \phi_s(\lambda x\, .\, \mathsf{GBR}^{\phi,q,\varphi}(s * x)) & \text{otherwise,} \end{cases} \tag{4.1}$$

where the parameters of GBR have types $\varphi \colon \rho^* \to ((\rho \to \tau) \to \tau)$, $q \colon \rho^* \to \tau$ and $\varphi \colon \rho^{\mathbb{N}} \to \mathbb{N}$. Note that the parameters in the superscript remain fixed in the defining axiom, in future these are often omitted when there is no risk of ambiguity.

Bar recursion is fundamentally different from primitive recursion. The primitive recursor R is well-defined by well-foundedness of the natural numbers: we assign a value to $\mathsf{R}(0)$ and define $\mathsf{R}(n+1)$ in terms of $\mathsf{R}(n)$. In contrast $\mathsf{GBR}(\langle\rangle)$ is well-defined by well-foundedness of the tree $T(s) := \forall n < |s| (\varphi(\overline{s,n}) \geq n)$: we assign a value $q(s)$ to $\mathsf{GBR}(s)$ whenever s is a leaf of T, and otherwise decide $\mathsf{GBR}(s)$ based on its value at extensions of s, or in other words in terms of the function $\lambda x\, .\, \mathsf{GBR}(s * x)$.

In the same way that primitive recursion is the computational analogue of the axiom of induction, Spector's bar recursion can be viewed as a computational analogue of the axiom of bar induction. There are several formulations of the principle of bar induction (which is of particular importance in intuitionistic mathematics, see e.g. [90]). Here we use the schema of relativised bar induction, given by

$$\mathsf{BI}_\rho \ : \ \left\{\begin{array}{c} S(\langle\rangle) \\ \wedge\forall\alpha \in S\exists n P([\alpha](n)) \\ \wedge\forall s \in S(\forall x(S(s*x) \to P(s*x)) \to P(s)) \end{array}\right\} \to P(\langle\rangle).$$

for P, S arbitrary formulas on ρ^*, where $\alpha \in S$ and $s \in S$ are shorthand for $\forall n S([\alpha](n))$ and $S(s)$ respectively. The axiom schema of relativised quantifier-free bar induction $\mathsf{QF\text{-}BI}$ is bar induction for P restricted to being quantifier-free. Note that bar induction is intuitionistically acceptable only with either a decidability or a monotonicity condition on S (otherwise it implies e.g. law of excluded middle for Π^0_1-formulas - see [42]). In most cases we will not need to relativise the bar induction at all, simply taking $S(s) = \mathsf{true}$.

Spector's bar recursion GBR is one of the most well known instances of recursion over well-founded trees, although it is just one of many extensions of system T in the literature, which also include *modified bar recursion* [11], the *Berardi-Bezem-Coquand functional* [8] and open recursion [10]. Extensions of system T and their relationship to one another will form the subject of Part III of this thesis.

4.1 Spector's condition

A necessary condition for Spector's bar recursion to exist in a model of $\mathsf{E\text{-}HA}^\omega$ is that the underlying tree over which the recursion is carried out is well-founded, or in more precise terms the following principle is satisfied:

$$\mathsf{Spec} \ : \ \forall\varphi^{\rho^{\mathbb{N}}\to\mathbb{N}}, \alpha^{\rho^{\mathbb{N}}}\exists n(\varphi(\overline{\alpha, n}) < n). \tag{4.2}$$

We refer to Spec as *Spector's condition.* In fact the least n satisfying this condition can be explicitly constructed in $\mathsf{E\text{-}HA}^\omega + \mathsf{GBR}$.

Proposition 4.1 (Howard [39][1]). *Define* $\theta_{\varphi,\alpha} \colon \rho^* \to \mathbb{N}$ *in* $\mathsf{E\text{-}HA}^\omega + \mathsf{GBR}$ *as*

$$\theta_{\varphi,\alpha}(s) := \begin{cases} 0 & \textit{if } \exists t \preceq s(\varphi(\hat{t}) < |t|) \\ 1 + \theta_{\varphi,\alpha}(s * \alpha(|s|)) & \textit{otherwise.} \end{cases}$$

Then provably in $\mathsf{E\text{-}HA}^\omega + \mathsf{GBR}$ *we have*

$$\exists n \leq \theta_{\varphi,\alpha}(\langle\rangle)(\varphi(\overline{\alpha, n}) < n),$$

[1] This is the standard reference, although Howard attributes the trick to Kreisel.

and therefore using bounded search can construct a term $\mu_{Sp}(\varphi, \alpha)$ in $\mathsf{E\text{-}HA}^\omega + \mathsf{GBR}$ *satisfying*

$$\mathsf{E\text{-}HA}^\omega + \mathsf{GBR} \vdash \varphi(\overline{\alpha, \mu_{Sp}(\varphi, \alpha)}) < \mu_{Sp}(\varphi, \alpha) \wedge (n < \mu_{Sp}(\varphi, \alpha) \to \varphi(\overline{\alpha, n}) \geq n).$$

Proof. Define $\beta i := \theta_{\varphi,\alpha}([\alpha](i))$. Then β satisfies

$$\beta i = \begin{cases} 0 & \text{if } \exists n \leq i(\varphi(\overline{\alpha, n}) < n) \\ 1 + \beta(i+1) & \text{otherwise.} \end{cases} \tag{4.3}$$

and by (4.3) we have

$$(\beta i \neq 0 \wedge j \leq i) \to \beta j = 1 + \beta(j+1) \tag{4.4}$$

and hence by induction from $j = 0$ to $j = i - 1$

$$\beta i \neq 0 \to \beta 0 = i + \beta i. \tag{4.5}$$

Setting $i = \beta 0$ it follows from (4.5) that $\beta(\beta 0) = 0$, and therefore by (4.3) we have

$$\exists n \leq \beta 0(\varphi(\overline{\alpha, n}) < n).$$

But $\beta 0 = \theta_{\varphi,\alpha}(\langle\rangle)$ so we're done. □

We call μ_{Sp} Spector's search functional, as it finds the least number satisfying Spector's condition. It has been pointed out by Oliva that the search functional μ_{Sp} is actually *definable*, though not validated, in $\mathsf{E\text{-}HA}^\omega$.

Proposition 4.2. *Define $\gamma_{\varphi,\alpha} \colon \rho^{\mathbb{N}}$ in* $\mathsf{E\text{-}HA}^\omega$ *by*

$$\gamma_{\varphi,\alpha}(i) := \begin{cases} 0_\rho & \text{if } \exists k \leq i + 1(\varphi(\overline{\alpha, k}) < k) \\ \alpha(i) & \text{otherwise,} \end{cases}$$

and define $\tilde{\mu}_{Sp}(\varphi, \alpha)$ as the least $n \leq \varphi(\gamma_{\varphi,\alpha}) + 1$ satisfying $\varphi(\overline{\alpha, n}) < n$, or just 0 if none exist. Then

$$\mathsf{E\text{-}HA}^\omega \vdash \varphi(\overline{\alpha, N}) < N \to \varphi(\overline{\alpha, \tilde{\mu}_{Sp}(\varphi, \alpha)}) < \tilde{\mu}_{Sp}(\varphi, \alpha) \wedge (n < \tilde{\mu}_{Sp}(\varphi, \alpha) \to \varphi(\overline{\alpha, n}) \geq n).$$

Proof. Assuming $\varphi(\overline{\alpha, N}) < N$ let $m \leq N$ be the least number satisfying Spector's condition. Then for $i < m - 1$ we have $\gamma_{\varphi,\alpha}(i) = \alpha(i)$, and for $i \geq m - 1$ we have $\gamma_{\varphi,\alpha}(i) = 0_\rho$, which implies that $\gamma_{\varphi,\alpha} = \overline{\alpha, m - 1}$. By minimality of m we must have $\varphi(\gamma_{\varphi,\alpha}) \geq m - 1$ and hence $m \leq \varphi(\gamma_{\varphi,\alpha}) + 1$, and the search bounded by $\tilde{\mu}_{\mathrm{Sp}}(\varphi, \alpha)$ must find m. □

4.2 Spector's computational interpretation of $\mathsf{AC}^{\mathbb{N}}$

Over intuitionistic logic the functional interpretation of the axiom of choice is trivial, but in the presence of classical logic things become problematic because $\mathsf{WE\text{-}PA}^{\omega} + \mathsf{QF\text{-}AC} + \mathsf{AC}^{\mathbb{N}}$ is no longer closed under the negative translation. However, $(\mathsf{AC}^{\mathbb{N}})^{\mathrm{N}}$, which is equivalent to

$$\forall n \neg\neg \exists x^{\rho} A_n(x)^{\mathrm{N}} \to \neg\neg \exists f \forall n (A_n(fn))^{\mathrm{N}}$$

is derivable in $\mathsf{WE\text{-}HA}^{\omega} + \mathsf{AC}^{\mathbb{N}} + \mathsf{DNS}$ where DNS is the intuitionistically unprovable *double negation shift*

$$\mathsf{DNS}\ :\ \forall n^{\mathbb{N}} \neg\neg B(n) \to \neg\neg \forall n B(n).$$

Suppose that $B(n)^D = \exists x^{\rho} \forall y^{\tau} |B(n)|^x_y$. Then the Dialectica interpretation of DNS is given by

$$\exists f, p, n \forall \varepsilon, q, \varphi \left(|B(n)|^{\varepsilon_n p}_{p(\varepsilon_n p)} \to |B(\varphi f)|^{f(\varphi f)}_{qf} \right), \tag{4.6}$$

where $f \colon \rho^{\mathbb{N}}$, $p \colon \rho \to \tau$ and n are dependent on $\varepsilon \colon \mathbb{N} \to ((\rho \to \tau) \to \rho)$, $q \colon \rho^{\mathbb{N}} \to \tau$ and $\varphi \colon \rho^{\mathbb{N}} \to \mathbb{N}$. Spector's achievement was to solve DNS^D using bar recursion. In fact, he used a restricted form SBR of GBR which we call his 'special' variant of bar recursion. The constant SBR_{ρ} for arbitrary ρ has defining axiom

$$\mathsf{SBR}^{\phi,\varphi}_{\rho}(s^{\rho^*}) \overset{\rho^{\mathbb{N}}}{:=} s \ @ \begin{cases} 0_{\rho^{\mathbb{N}}} & \text{if } \varphi(\hat{s}) < |s| \\ \mathsf{SBR}^{\phi,\varphi}_{\rho}(s * a_s) & \text{otherwise} \end{cases} \tag{4.7}$$

where $a_s := \phi_s(\lambda x \,.\, \mathsf{SBR}(s * x))$, and the parameters have types $\phi \colon \rho^* \to ((\rho \to \rho^{\mathbb{N}}) \to \rho)$, $\varphi \colon \rho^{\mathbb{N}} \to \mathbb{N}$ respectively.

Theorem 4.3 (Spector [86]). *Given ε, q and φ as above, define $f \colon \rho^{\mathbb{N}}$ and $p_n \colon \rho \to \tau$ by*

$$\begin{aligned} f &:= \mathsf{SBR}^{\phi^{\varepsilon},\varphi}_{\rho}(\langle\rangle) \\ p_n &:= \lambda x \,.\, q(\mathsf{SBR}^{\phi^{\varepsilon},\varphi}_{\rho}([f](n))) \end{aligned}$$

where $\phi^{\varepsilon}(P^{\rho \to \rho^{\mathbb{N}}}) := \varepsilon_{|s|}(\lambda x \,.\, q(P(x)))$. Then f and p_n satisfy

$$\begin{aligned} fn &= \varepsilon_n p_n \\ p_n(fn) &= qf \end{aligned} \tag{4.8}$$

for all $n \leq \varphi f$, and therefore in particular f, $p := p_{\varphi f}$ and $n := \varphi f$ witness the Dialectica interpretation of DNS *(4.6).*

We prove Spector's theorem under the guise of the product of selection functions in Theorem 5.7, and cover its extension to dependent choice (which first appears in [61]) in Theorem 5.13. A corollary of Theorem 4.3 is soundness of the functional interpretation for classical analysis.

Theorem 4.4. *Let $A(a)$ be a formula in $\mathcal{L}(\mathsf{WE\text{-}PA}^\omega)$ containing only a free. Then*

$$\mathsf{WE\text{-}PA}^\omega + \mathsf{QF\text{-}AC} + \mathsf{AC}^{\mathbb{N}} \vdash A(a) \Rightarrow \mathsf{WE\text{-}HA}^\omega + \mathsf{SBR} \vdash \forall y |A(a)^{\mathrm{N}}|_y^{t(a)}$$

where t is a closed term of $\mathsf{WE\text{-}HA}^\omega + \mathsf{SBR}$ *which can be extracted from the proof of A.*

Remark 4.5. By $\mathsf{WE\text{-}HA}^\omega + \mathsf{SBR}$ we of course mean $\mathsf{WE\text{-}HA}^\omega$ extended with constants SBR_ρ of each type along with their defining axioms.

Remark 4.6. That the conclusion of Theorem 4.4 can be verified in $\mathsf{WE\text{-}HA}^\omega + \mathsf{SBR}$ does not in fact follow from Spector, who uses higher-type extensionality to validate his realizer. This is pointed out in [61] where it is claimed that a constructive ω-rule is required, but in fact a precise formalization of Spector's proof is only given in [48, p.204], where it is shown that a quantifier-free fragment of $\mathsf{WE\text{-}HA}^\omega + \mathsf{SBR}$ is sufficient for the verification.

4.3 Howard's computational interpretation of WKL

In [41] Howard demonstrated that a simple, binary form of bar recursion was sufficient to realize the ND interpretation of weak König's lemma (see Section 2.3). We make use of Howard's realizer in Part II, so we state and prove his result here. Under the assumption that T is a binary tree (i.e. satisfying $T(s * t) \to T(s)$) can state WKL as

$$\forall n T(hn) \to \exists \alpha \forall n T([\alpha](n))$$

where to eliminate the bounded quantifier $\exists s \in \mathbb{B}^n$ we have defined h as

$$hn := \text{least code of an } s \in \mathbb{B}^n \text{ satisfying } T(s) \text{ , } 0_{\mathbb{B}^n} \text{ if none exist}$$

for a suitable coding of $\mathbb{B}^n$. The negative translation of WKL is then equivalent to

$$\forall n T(hn) \to \neg\neg\exists \alpha \forall n T([\alpha](n)),$$

and therefore its functional interpretation is given by

$$\exists \alpha, n \forall \varphi \left(T(hn) \to T([\alpha](\varphi\alpha)) \right). \tag{4.9}$$

where α and n are now dependent on $\varphi\colon \mathbb{B}^{\mathbb{N}} \to \mathbb{N}$.

Theorem 4.7 (Howard [40]). *Define $K_\varphi\colon \mathbb{B}^* \to \mathbb{N}$ using $\mathsf{GBR}_{\mathbb{B},\mathbb{N}}$ as*

$$K_\varphi(s) := \begin{cases} 0 & \text{if } \varphi(\hat{s}) < |s| \\ 1 + \max\{K_\varphi(s * 0), K_\varphi(s * 1)\} & \text{otherwise.} \end{cases} \tag{4.10}$$

Then $N_\varphi := K_\varphi(\langle\rangle)$ and $\alpha_\varphi := \hat{u}$ for least $u \preceq hN_\varphi$ satisfying $\varphi(\hat{u}) < |u|$ (or $\langle\rangle$ otherwise) satisfy the ND interpretation of WKL *(4.9).*

Actually, in his original paper [40] Howard proves that $n := K_\varphi(\langle\rangle)$ witnesses the contrapositive of WKL in the form of the FAN principle:

$$\forall\varphi(\forall\alpha\neg T([\alpha](\varphi\alpha)) \to \neg T(h(K_\varphi(\langle\rangle)))).$$

His proof uses bar induction and Proposition 4.1, and is outlined in considerable detail in [77]. We give a slightly shorter verification that his realizer witnesses the ND interpretation of WKL.

Proof of Theorem 4.7. We begin by proving that $K_\varphi(\langle\rangle)$ is a uniform bound on the first point at which binary sequences satisfy Spector's condition:

$$\forall s(|s| \geq K_\varphi(\langle\rangle) \to \exists n \leq |s|(\varphi(\overline{s,n}) < n)).$$

For arbitrary s, suppose that

$$(*) \;\; \forall n \leq |s|(\varphi(\overline{s,n}) \geq n).$$

Then for $n < |s|$ we have

$$K_\varphi([s](n)) \overset{(*)}{=} 1 + \max\{K_\varphi([s](n) * 0), K_\varphi([s](n) * 1)\} \geq 1 + K_\varphi([s](n+1)).$$

By induction from $n = 0$ to $n = |s| - 1$ we obtain

$$K_\varphi(\langle\rangle) \geq |s| + K_\varphi(s),$$

and by $(*)$ for $n = |s|$ we have $K_\varphi(s) > 0$ and therefore

$$K_\varphi(\langle\rangle) > |s|.$$

This proves that $K_\varphi(\langle\rangle)$ is a uniform bound, so setting $N_\varphi = K_\varphi(\langle\rangle)$ we know that there is some $u \preceq h(N_\varphi)$ satisfying $\varphi(\widehat{u}) < |u| \leq N_\varphi$. Setting $\alpha_\varphi = \widehat{u}$ we have

$$[\alpha_\varphi](\varphi(\alpha_\varphi)) = [\widehat{u}](\varphi(\widehat{u})) \overset{\varphi(\widehat{u})<|u|}{=} [u](\varphi(\widehat{u})) \prec u \preceq h(N_\varphi)$$

and therefore

$$T(h(N_\varphi)) \to T([\alpha_\varphi](\varphi(\alpha_\varphi)))$$

by prefix closure of T, which is the ND interpretation of WKL. □

Remark 4.8. Note that technically the full statement of weak König's lemma must explicitly include the assumption that the predicate T is a binary tree i.e. $\forall s, t(T(s*t) \to T(s))$, and therefore its functional interpretation must calibrate exactly how, for each φ, we use the tree property. We ignore the assumption here at it has no computational content, and a quick inspection of preceding proof confirms that our realizer only requires closure for prefixes of the branch hN_φ.

Corollary 4.9. *Let $A(a)$ be a formula in $\mathcal{L}(\mathsf{WE\text{-}PA}^\omega)$ containing only a free. Then*

$$\mathsf{WE\text{-}PA}^\omega + \mathsf{QF\text{-}AC} + \mathsf{WKL} \vdash A(a) \Rightarrow \mathsf{WE\text{-}HA}^\omega + \mathsf{GBR}_{\mathbb{B},\mathbb{N}} \vdash \forall y |A(a)^{\mathrm{N}}|^{t(a)}_{y}$$

where t is a closed term of $\mathsf{WE\text{-}HA}^\omega + \mathsf{GBR}_{\mathbb{B},\mathbb{N}}$ which can be extracted from the proof of A.

4.4 Models of GBR

An immediate corollary of the fact that $\mathsf{E\text{-}HA}^\omega + \mathsf{GBR}$ validates Spector's condition (4.2) is that the type structure of full set theoretic functionals $\mathscr{S}^\omega$ is *not* a model of bar recursion. Define $\varphi\colon \mathbb{N}^\mathbb{N} \to \mathbb{N}$ by

$$\varphi(\alpha) := i + 1 \text{ for least } i \text{ such that } \alpha i = 0 \text{ if it exists, else } 0.$$

Then Spector's condition fails for φ and $\alpha = \lambda i.1$ since $\varphi(\overline{\alpha, n}) = n + 1$ for all n. This confirms that the theory $\mathsf{E\text{-}HA}^\omega + \mathsf{GBR}$ is a proper extension of $\mathsf{E\text{-}HA}^\omega$ because $\mathscr{S}^\omega$ *is* a model of $\mathsf{E\text{-}HA}^\omega$.

However, there are several important structures that do validate bar recursion.

4.4.1 Term models

Term models for bar recursion were first developed by Luckhardt [61] and Tait [87]. Strong normalisation for bar recursive functionals without using infinite terms is proved in [14].

4.4.2 The continuous functionals

There are several equivalent ways of formulating the type stucture of continuous functionals $\mathscr{C}^\omega$, the most well-known being those of Kleene [45] (via associates) and Kreisel [54] (via formal neighbourhoods). We direct the reader to e.g. Norman [65] for details of these structures. It is a standard result that the continuous functionals are a model of bar recursion.

Theorem 4.10. *$\mathscr{C}^\omega$ is a model of* $\mathsf{E\text{-}HA}^\omega + \mathsf{GBR}$.

Theorem 4.10 was first proved by Scarpellini [79], who uses yet another formulation of the continuous functionals, the so-called *sequentially continuous functionals* (which are nevertheless shown to be equivalent to the usual Kleene continuous functionals in [43]). One can alternatively justify bar recursion in the continuous functionals by appealing to Ershov's result [26] that $\mathscr{C}^\omega$ can be identified with the total elements of the model $\hat{\mathscr{C}}^\omega$ of partial continuous functionals - see [11].

Informally, Spector's condition holds in $\mathscr{C}^\omega$ because it validates the axiom of *continuity*:

$$\mathsf{Cont} \ : \ \forall F^{\rho^\mathbb{N} \to \tau}, \alpha^{\rho^\mathbb{N}} \exists N \forall \beta^{\rho^\mathbb{N}} ([\alpha](N) =_{\rho^*} [\beta](N) \to F(\alpha) =_\tau F(\beta)),$$

where the type τ is restricted to being discrete. To see this, note that for any $\varphi\colon \rho^\mathbb{N} \to \mathbb{N}$ and $\alpha\colon \rho^\mathbb{N}$ there exists, by Cont, a number N such that $\varphi(\overline{\alpha, n}) = \varphi(\alpha)$ for all $n \geq N$. Setting $n = \max\{N, \varphi(\alpha) + 1\}$ we have

$$\varphi(\overline{\alpha, n}) = \varphi(\alpha) < n.$$

4.4.3 The majorizable functionals

For many years bar recursion was only considered in the context of the continuous functionals. However, in 1985 Bezem proved the remarkable fact that the strongly majorizable functionals are a model of GBR, despite the fact that they include discontinuous functionals.

The relation *s-maj* is defined inductively on the finite types by

$$\begin{cases} n^* \ s\text{-}maj_{\mathbb{N}} \ n :\equiv n^* \geq n \\ f^* \ s\text{-}maj_{\rho\to\tau} \ f :\equiv \forall x^*, x(x^* \ s\text{-}maj_\rho \ x \to f^*x^* \ s\text{-}maj_\tau \ f^*x, fx). \end{cases}$$

(we omit details of how one could extend it to our enriched system of types). We say that $x\colon \rho$ is strongly majorizable if there exists some x^* satisfying x^* $s\text{-}maj_\rho$ x, and call x^* a majorant of x. The type structure $\mathscr{M}^\omega := \langle M_\rho \rangle$ of hereditarily strongly majorizable functionals is defined inductively by $M_0 := \mathbb{N}$, and $M_{\rho\to\tau}$ is the space of all set-theoretic functionals $M_\tau^{M_\rho}$ which also have a majorant in $M_\tau^{M_\rho}$ (see [15] or [48] for details).

Theorem 4.11 ([15])**.** *$\mathscr{M}^\omega$ is a model of* $\mathsf{E\text{-}HA}^\omega + \mathsf{GBR}$.

While the full proof of this result is rather technical, to see that Spector's condition holds in $\mathscr{M}^\omega$, suppose that φ^* and α^* are majorants of α and φ respectively. Then α^* is also a majorant of $\overline{\alpha, n}$ for arbitrary n, so setting $n = \varphi^*\alpha^* + 1$ we have $\varphi^*\alpha^*$ $s\text{-}maj_{\mathbb{N}}$ $\varphi(\overline{\alpha, n})$ i.e.

$$\varphi(\overline{\alpha, n}) \leq \varphi^*\alpha^* < n.$$

II The Constructive Content of Mathematical Analysis

CHAPTER

FIVE

The Semantics of Bar Recursive Functionals

The first main contribution of this dissertation will be to present a series of case studies in which we use Gödel's functional interpretation to extract bar recursive programs from proofs in classical analysis. Rather than merely stating the extracted programs in computer code or abstruse logical syntax, our goal is to study the meaning and behaviour of these rather complicated functionals so that they can be understood on a *mathematical* level as intuitive constructive analogues to non-constructive principles. Our primary tool for interpreting choice principles will be Spector's bar recursion in the form of the product of selection functions. However, as part of our analysis we address the issue of whether or not bar recursion is always best suited for obtaining illuminating programs from proofs, and consider the possibility that alternatives to bar recursion, or at least new descriptions of bar recursive realizers, might be more appropriate when interpreting certain non-constructive ideas. In broad terms our aim is that these case studies will provide some insight into the constructive nature of classical analysis. In doing so we also hope to highlight the functional interpretation as an elegant translation on proofs whose output can be read and understood on mathematical terms - a feature that can often become obscured beneath the formal language of logic.

* * *

Realizers extracted from a classical proofs using proof interpretations can yield highly non-trivial constructive information hidden deep in the logical structure of that proof. This feature has been exploited in the proof mining program to obtain explicit numerical information from non-constructive proofs, which has let to improved results in several areas of mathematics.

However, realizing terms give us more that just numerical information. A functional extracted from a non-constructive proof is ultimately a constructive reformulation of that proof, and an analysis of the behaviour of that functional can reveal insights into the computational semantics of the proof. In recent years there has been an increasing interest in applying proof theoretic techniques for the sole purpose of understanding the constructive meaning of non-constructive principles, whether it is to explore a specific non-constructive idea (such as the analysis of the minimal bad sequence argument in [82]) or to give classical logic a general computational semantics (such as the learning-based realizability of [3]). In many ways work of this kind forms a natural continuation of Hilbert's program, and while (outside of the proof mining program) the direct impact of this research on mainstream mathematics may appear limited, there are complex links between proof theory and mathematics - such as that between the functional interpretation and the correspondence principle described in Section 3.3 - that lend force to the idea that understanding the constructive meaning of non-constructive ideas has a relevance beyond the world of computer science and logic.

Extracting computational content from proofs can be a difficult task. The soundness theorems for the functional interpretation guarantee that we can extract realizers from a large class of proofs and come equipped with an algorithm for doing so providing the proof has been fully formalised, and in this sense program extraction is mathematically trivial: it is a procedure that could be automated by a theorem prover. But aside even from the laborious task of fully formalising a non-trivial proof, such a blind implementation would yield little more than a 'black box', a complex higher-type term that mysteriously works but provides little immediate insight into the proof it interprets. In reality, it often requires a certain amount of work and the application of new ideas or techniques to obtain constructive information that is useful or meaningful.

In this practical part of this dissertation our challenge is to use the functional interpretation to analyse the constructive content of several important principles in mathematical analysis, with the aim of obtaining programs whose behaviour and semantics can be clearly understood. Our starting point is the recent work of Escardó and Oliva [29, 30, 31], who have shown that Spector's bar recursion is closely related to the so-called 'explicitly iterated' product of selection functions. This form of recursion can be intuitively viewed as an operation that computes optimal plays in a class of unbounded sequential games, and therefore realizers extracted using the product of selection functions come equipped with a natural game semantics. We focus on giving a detailed description of exactly what our realizers are attempting to do, and how their behaviour can be related to the non-constructive ideas present in the original proof. In certain cases we suggest alternative ways of viewing and constructing these realizers.

We have deliberately chosen to study well-known proofs in mathematics, and in what follows we are generally not the first to analyse the computational content of these proofs. The originality of work is that we give new realizers and more importantly new descriptions of these realizers. However, in Chapter 8 we give an analysis of minimal-bad-sequence argument for arbitrary well-founded relations via transfinite recursion, which to our knowledge constitutes one of the first applications of the functional interpretation in the theory of well-quasi-orderings.

We adhere to an informal style of presentation and whenever possible write in a familiar mathematical language. We make no attempt to give full details of how our realizers can be formally obtained from proofs - in all cases they are extracted 'by hand', and this contrasts from the more rigorous, syntactic approach often found elsewhere in the literature.

The remainder of this chapter is dedicated to providing technical background and outlining the main ideas that will be used in the case studies that follow.

5.1 The explicitly iterated product of selection functions

We first introduce the product of selection functions and show that it can be viewed as a functional that computes optimal strategies in unbounded sequential games. This section is a survey of results due to Escardó and Oliva, which forms part of a larger body of research by these authors on selection functions and game theory. In particular the reader is referred to the original paper [30] and the more recent [31] for general background to the work presented here.

A *selection function* is any element of type $(\rho \to \tau) \to \rho$. As in [30] we abbreviate this type to $J_\tau \rho$. Closely related to the selection function $\varepsilon \colon J_\tau \rho$ is its corresponding *quantifier* $\bar{\varepsilon} \colon (\rho \to \tau) \to \tau$ defined by $\bar{\varepsilon} p := p(\varepsilon p)$. The intuition is to view ε as a selector that given a function $p \colon \rho \to \tau$ picks a particular element of εp of ρ that *attains* its quantifier $\bar{\varepsilon}$. Selection functions and quantifiers are ubiquitous in mathematics and computer science, as the following examples illustrate.

Example 5.1. (a) A canonical example of a selection function and its associated quantifier arises when τ forms a set of truth values e.g. $\tau = \mathbb{B}$. Hilbert's epsilon term $\varepsilon_\rho \colon J_{\mathbb{B}} \rho$ of type ρ is a selection function whose corresponding quantifier is just the usual existential quantifier $\exists_\rho$ for predicates over type ρ, since classically we have

$$\exists x^\rho \, p(x) \Leftrightarrow p(\varepsilon_\rho p).$$

(b) By the mean value theorem there exists a selection function $\varepsilon \colon J_{[0,1]} \mathbb{R}$ such that for any continuous function $p \colon [0,1] \to \mathbb{R}$ we have

$$p(\varepsilon p) = \int_0^1 p(x) dx.$$

Its corresponding quantifier is the operator $\int_0^1$.

(c) Assume we are given a position in a game where we have to pick a move in ρ. A *strategy* for that position can be defined by a selection function $\varepsilon \colon J_\tau \rho$ that determines an optimal move for this position, given a mapping $p\colon \rho \to \tau$ of possible moves $x\colon \rho$ to corresponding outcomes $p(x)\colon \tau$.

There is a natural product operation $\otimes$ with which one can compose selection functions.

Definition 5.2 (Binary product of selection functions [30]). Given a selection function $\varepsilon\colon J_\tau\rho$, a family of selection functions $\delta\colon \rho \to J_\tau\sigma$ and a function $q\colon \rho \times \sigma \to \tau$, let

$$\begin{aligned} A[x^\rho] &\overset{\sigma}{:=} \delta_x(\lambda y.q(x,y)) \\ a &\overset{\rho}{:=} \varepsilon(\lambda x.q(x, A[x])). \end{aligned}$$

The binary product $\varepsilon \otimes \delta$ is another selection function, of type $J_\tau(\rho \times \sigma)$, defined by

$$(\varepsilon \otimes \delta)(q) \overset{\rho\times\sigma}{:=} \langle a, A[a]\rangle.$$

If δ is independent of x we call this the *simple* product of selection functions. The general case is then sometimes referred to as the *dependent* product of selection functions.

Example 5.3. Continuing from Example 5.1:

(a) It is easy to show that the product of ε operators $\varepsilon_\rho \otimes \varepsilon_\sigma$ is an epsilon operator of type $\rho \times \sigma$ in the sense that

$$\exists x^\rho \exists y^\sigma q(x,y) \Leftrightarrow q((\varepsilon_\rho \otimes \varepsilon_\sigma)(q)).$$

(b) Given a continuous function $q\colon [0,1]^2 \to \mathbb{R}$ we have

$$q((\varepsilon \otimes \varepsilon)(q)) = \int_0^1 \int_0^1 q(x,y)\; dxdy.$$

(c) Given strategies ε_0, ε_1 for each round in a two round sequential game with outcome function $q\colon \rho_0 \times \rho_1 \to \tau$, then $(\varepsilon_0 \otimes \varepsilon_1)(q)$ forms a *strategy* for the game which is “compatible” with the local strategies ε_0 and ε_1. This key instance of the product is discussed in more detail in Section 5.2.

Remark 5.4. The binary product also appears throughout applied proof theory, due to its connection with the functional interpretation of finite choice principles (as elaborated in Section 5.3). For example, it is implicitly used in e.g. [50, p.14].

The *explicitly iterated* product of selection functions EPS is an infinite iteration of the binary product $\otimes$, controlled by a functional φ which stops the iteration once Spector's condition (4.2) is obtained. Note that the following definition differs from that given in e.g. [31, Definition 15] (in that case the parameter q changes with each recursive call, whereas we prefer to minimise the number of parameters that change as part of the defining axiom).

Definition 5.5 (Explicitly iterated product of selection functions). The explicitly iterated product of selection functions $\mathsf{EPS}_{\rho,\tau}$ has defining equation

$$\mathsf{EPS}^{\varepsilon,q,\varphi}(s^{\rho^*}) \overset{\rho^{\mathbb{N}}}{=} s \mathbin{@} \begin{cases} 0_{\rho^{\mathbb{N}}} & \text{if } \varphi(\hat{s}) < |s| \\ \\ \mathsf{EPS}^{\varepsilon,q,\varphi}(s * a_s) & \text{otherwise,} \end{cases} \tag{5.1}$$

for $a_s := \varepsilon_s(\lambda x \,.\, q(\mathsf{EPS}^{\varepsilon,q,\varphi}(s * x)))$, where $\varepsilon \colon \rho^* \to J_\tau \rho$ is a family of selection functions, $q \colon \rho^{\mathbb{N}} \to \tau$ is referred to as the *outcome* functional and $\varphi \colon \rho^{\mathbb{N}} \to \mathbb{N}$ is referred to as the *control* functional.

Note that EPS is very similar to Spector's special form of bar recursion SBR (4.7), the key difference being the introduction of the outcome functional q. In fact, it is easy to see that SBR is precisely the instance of EPS in which $\tau = \rho^{\mathbb{N}}$ and $q \colon \rho^{\mathbb{N}} \to \rho^{\mathbb{N}}$ is the identity functional. Conversely, EPS is definable from SBR: if we set $\tilde{\varepsilon}^q_s(p^{\rho \to \rho^{\mathbb{N}}}) \overset{\rho}{=} \varepsilon_s(\lambda x \,.\, q(px))$ then $\mathsf{SBR}^{\tilde{\varepsilon}^q,\varphi}$ satisfies the defining equation of $\mathsf{EPS}^{\varepsilon,q,\varphi}$. EPS is precisely what we need to solve Spector's equations (4.8), a fact that we now prove in full. In what follows we suppress the parameters of EPS when it is clear what we mean.

Lemma 5.6. *Let $\alpha = \mathsf{EPS}^{\varepsilon,q,\varphi}(s)$. Then for all $n \geq |s|$,*

$$\alpha = \mathsf{EPS}^{\varepsilon,q,\varphi}([\alpha](n)). \tag{5.2}$$

Proof. Induction on n. For $n = |s|$ we have $[\alpha](n) = [\mathsf{EPS}(s)](|s|) = s$ and therefore (5.2) holds by definition. Now assume (5.2) holds for $n \geq |s|$. There are two cases:

(a) If $\varphi(\widehat{[\alpha](n)}) = \varphi(\overline{\alpha, n}) < n$ then

$$\alpha \overset{\text{(IH)}}{=} \mathsf{EPS}([\alpha](n)) = [\alpha](n) \mathbin{@} 0 \equiv \overline{\alpha, n}.$$

But then $\varphi(\widehat{[\alpha](n+1)}) = \varphi(\overline{\alpha, n+1}) = \varphi(\overline{\alpha, n}) < n < n+1$ by extensionality, and therefore $\mathsf{EPS}([\alpha](n+1)) = [\alpha](n+1) \mathbin{@} 0 = \overline{\alpha, n+1} = \overline{\alpha, n} = \alpha$.

(b) Otherwise,

$$\alpha \overset{\text{(IH)}}{=} \mathsf{EPS}([\alpha](n)) = [\alpha](n) \mathbin{@} \mathsf{EPS}([\alpha](n) * a) \overset{(*)}{=} \mathsf{EPS}([\alpha](n) * a)$$

where $a = \varepsilon_{[\alpha](n)}(\lambda x \,.\, q(\mathsf{EPS}([\alpha](n) * x)))$. Note that $(*)$ holds because $[\alpha](n)$ is a prefix of $\mathsf{EPS}([\alpha](n) * a)$. Now then $\alpha(n) = \mathsf{EPS}([\alpha](n) * a)(n) = a$ and therefore $\alpha = \mathsf{EPS}([\alpha](n) * \alpha(n)) = \mathsf{EPS}([\alpha](n+1))$. □

This lemma is the building block behind the following fundamental theorem on EPS.

Theorem 5.7. *Define $\alpha \colon \rho^{\mathbb{N}}$ and $p_s \colon \rho \to \tau$ by*

$$\begin{aligned} \alpha &:= \mathsf{EPS}^{\varepsilon,q,\varphi}_{\rho,\tau}(\langle\rangle) \\ p_s &:= \lambda x \,.\, q(\mathsf{EPS}^{\varepsilon,q,\varphi}_{\rho,\tau}(s * x)). \end{aligned}$$

Then for all $n \leq \varphi\alpha$ we have

$$\begin{aligned} \alpha(n) &= \varepsilon_{[\alpha](n)}(p_{[\alpha](n)}) \\ p_{[\alpha](n)}(\alpha(n)) &= q\alpha. \end{aligned} \tag{5.3}$$

Proof. If $n \leq \varphi\alpha$ then we must have $(*)$ $\varphi(\widehat{[\alpha](n)}) \geq n$. Otherwise, if $\varphi(\widehat{[\alpha](n)}) < n$ then by Lemma 5.6 (for $s = \langle\rangle$) we would have $\alpha = \mathsf{EPS}([\alpha](n)) = [\alpha](n) \mathbin{@} \mathbf{0} = \widehat{[\alpha](n)}$, and hence $n \leq \varphi\alpha = \varphi(\widehat{[\alpha](n)}) < n$ which is a contradiction. Therefore

$$\begin{aligned} \alpha(n) &\overset{L5.6}{=} \mathsf{EPS}([\alpha](n))(n) \\ &\overset{(*)+(5.1)}{=} ([\alpha](n) \mathbin{@} \mathsf{EPS}([\alpha](n) * a_{[\alpha](n)}))(n) \\ &\overset{(5.1)}{=} a_{[\alpha](n)} \end{aligned}$$

and $a_{[\alpha](n)} = \varepsilon_{[\alpha](n)}(\lambda x \,.\, q(\mathsf{EPS}([\alpha](n) * x))) = \varepsilon_{[\alpha](n)}(p_{[\alpha](n)})$. For the second equality, we have

$$\begin{aligned} q\alpha &\overset{L5.6}{=} q(\mathsf{EPS}([\alpha](n+1))) \\ &= q(\mathsf{EPS}([\alpha](n) * \alpha(n))) \\ &= p_{[\alpha](n)}(\alpha(n)). \end{aligned}$$

□

Note that (5.3) is the dependent formulation of Spector's equations (4.8). The significance of the work done in [30, 31] regarding the product of selection functions lies in the observation that Spector's equations have a fundamental meaning that goes beyond the functional interpretation of choice, and that this meaning is elegantly expressed in terms of selection functions. In short, (5.3) characterise the product EPS as an operation that computes a kind of sequential equilibrium between selection functions. Such equilibria appear naturally in a variety of contexts, and we now outline perhaps the most illuminating of these contexts: the theory of sequential games.

5.2 Sequential games and optimal strategies

One of the most remarkable property of EPS is that it computes optimal strategies in a certain class of sequential games. The reader is encouraged to consult [31] in conjunction with the relatively concise discussion here (where in particular they discuss a more general class of games).

Definition 5.8 (Sequential game of unbounded length). A sequential game $\mathcal{G}$ is defined by a pair of types (ρ, τ) and a tuple of terms $(\varepsilon, q, \varphi)$ described as follows.

(a) ρ is the type of possible moves at each round,

(b) τ is the type of possible outcomes of the game,

(c) a finite sequence $s\colon \rho^*$ is thought of as a *position* in the game in representing moves for the first $|s|$ rounds, and an infinite sequence $\alpha\colon \rho^{\mathbb{N}}$ is considered to be a *play* of the game,

(d) $q\colon \rho^{\mathbb{N}} \to \tau$ determines the outcome of any given play α,

(e) $\varphi\colon \rho^{\mathbb{N}} \to \mathbb{N}$ determines the *relevant part* of a play,

(f) $\varepsilon\colon \rho^* \to J_\tau \rho$ defines a *strategy* for the game, where ε_s determines the optimal move at position s given that the outcome of all potential moves $\rho \to \tau$ is known.

Given a play α, all moves $\alpha(n)$ for $n \leq \varphi\alpha$ are *relevant moves.* A position s is called relevant if $|s| \leq \varphi(\hat{s})$, i.e. if in a canonical extension of the current position s the current move is considered a relevant move. We shall only consider infinite plays which are obtained by some canonical extension of a finite play s. Therefore, we think of these as finite games of unbounded length.

A *strategy* for the game is a function $\mathit{next} : \rho^* \to \rho$ which determines for each position s what the next move $\mathit{next}(s)$ should be. *To follow a strategy* from position s means to play all following moves according to the strategy, i.e. we obtain a sequence of moves $\alpha(0), \alpha(1), \ldots$ as

$$\alpha(n) = \mathit{next}(s * [\alpha](n)).$$

We call this the *strategic extension of s.* The strategic extension of the empty play is called the *strategic play.*

Definition 5.9. Given a game $\mathcal{G} = (\varepsilon, q, \varphi)$, a strategy *next* is said to be *optimal* if the move played at each relevant position s is the one recommended by the selection function ε_s i.e.

$$\mathit{next}(s) = \varepsilon_s(\lambda x \,.\, q(s * x * \alpha))$$

where α is the strategic extension of $s * x$. The strategic extension of the empty play when our strategy is optimal is the *optimal play* of the game $\mathcal{G}$.

The main result of [31] is that the product of selection functions computes optimal strategies.

Theorem 5.10 ([31]). *Given a game $\mathcal{G} = (\varepsilon, q, \varphi)$, the strategy*

$$\mathit{next}(s) = \mathsf{EPS}^{\varepsilon,q,\varphi}(s)(|s|) \tag{5.4}$$

is optimal, and, moreover,

$$\alpha = \lambda n \,.\, \mathsf{EPS}^{\varepsilon,q,\varphi}(s)(|s| + n) \tag{5.5}$$

is the strategic extension of s, i.e. $\alpha(n) = \mathit{next}(s * [\alpha](n))$. In particular $\mathsf{EPS}^{\varepsilon,q,\varphi}(\langle\rangle)$ is the optimal play of $\mathcal{G}$.

Proof. We have that

$$\begin{aligned}\alpha(n) &\overset{(5.5)}{=} \mathsf{EPS}(s)(|s|+n)\\ &\overset{L5.6}{=} \mathsf{EPS}(s * [\alpha](n))(|s|+n)\\ &\overset{(5.4)}{=} next(s * [\alpha](n)),\end{aligned}$$

which proves the second claim (5.5). Note that the second equality follows from Lemma 5.6 because $s * \alpha = \mathsf{EPS}(s)$ and since $|s| + n \geq |s|$ we have $s * \alpha = \mathsf{EPS}([s * \alpha](|s|+n)) = \mathsf{EPS}(s * [\alpha](n))$. Hence, assuming that s is a relevant position, i.e. $(*)$ $\varphi(\widehat{s}) \geq |s|$ we have

$$\begin{aligned}next(s) &\overset{(5.4)}{=} \mathsf{EPS}(s)(|s|)\\ &\overset{(*)+(5.1)}{=} (s \mathbin{@} \mathsf{EPS}(s * a_s))(|s|)\\ &= a_s\\ &= \varepsilon_s(\lambda x \,.\, q(\mathsf{EPS}(s * x)))\\ &= \varepsilon_s(\lambda x \,.\, q(s * x * \beta))\end{aligned}$$

where $\beta := \lambda n \,.\, \mathsf{EPS}(s * x)(|s|+1+n)$, and by the second claim (5.5) is the strategic extension of $s * x$. □

5.3 A game-theoretic computational interpretation of DC

We now prove that the product of selection functions EPS give a computational interpretation to dependent choice DC. This extends slightly the result from [30, 31] which is restricted to the interpretation of countable choice $\mathsf{AC}^{\mathbb{N}}$. In particular we define a variant of DC that is very naturally interpreted by EPS. We also attempt to explain in detail the link between the theory of selection functions and Gödel's functional interpretation - in particular the notion that optimal plays in a sequential games are somehow related to the axiom of choice.

Suppose we are given a Σ_2-theorem $A \equiv \exists x^\rho \forall y^\tau A_0(x,y)$ where A_0 is quantifier-free. The negative translation of A is equivalent to $\neg\neg\exists x \forall y A_0(x,y)$, and therefore its functional interpretation is given by

$$|A^{\mathrm{N}}|^{\varepsilon}_{p} = A_0(\varepsilon p, p(\varepsilon p)).$$

In other words, the functional interpretation eliminates double negations in front of a Σ_2-formula with a selection function $\varepsilon \colon J_\tau \rho$. As discussed in section 3.3, p can be viewed as a counterexample function attempting to disprove A, and therefore the constructive interpretation of A is precisely a selection function ε that refutes arbitrary counterexample functions.

Thus under the functional interpretation we have the following mapping:

$$\mathbf{\Sigma_2\text{-Theorems}} \quad \mapsto \quad \mathbf{Selection\ functions.}$$

The elimination of double negations in A^{N} for an arbitrary formula A is essentially a (albeit complex) modular iteration of this process, suggesting that selection functions and modes of recursion based on selection functions lie behind the functional interpretation in a fundamental way. Most importantly, the product of selection functions directly interprets the negative translation of the axiom of countable dependent choice.

For simplicity, let's first consider AC for Π_1-formulas:

$$\Pi_1\text{-}\mathsf{AC}\ :\ \forall n \exists x^\rho \forall y^\tau A_n(x,y) \to \exists f^{\mathbb{N}\to\rho} \forall n, y A_n(fn, y)$$

where the A_n are quantifier-free. Its negative translation is intuitionistically equivalent to

$$\forall n \neg\neg \exists x \forall y A_n(x,y) \to \neg\neg \exists f \forall n, y A_n(fn,y),$$

and the (partial) functional interpretation of this is given by

$$\exists F \forall \varepsilon, q, \varphi (\forall n, p A_n(\varepsilon_n p, p(\varepsilon_n p)) \to A_i(F(\varphi F), qF)). \tag{5.6}$$

This constructive interpretation of AC asks for a function $F^{\varepsilon,q,\varphi}$ approximating the choice sequence f, given a sequence of selection functions ε_n which interpret each of the A_n. By Theorem 5.7, $F^{\varepsilon,q,\varphi} := \mathsf{EPS}^{\varepsilon,q,\varphi}(\langle\rangle)$ does the job, indicating that under the functional interpretation we have a mapping

Choice principles $\mapsto$ Product of selection functions.

At first glance it may seem strange that an operation that computes optimal strategies in sequential games is related to the axiom of choice is this manner. But if we take a closer look, the game theoretic semantics of (5.6) becomes clear. The selection functions ε_n which realise the premise of (5.6) can be seen as a collection of strategies separately witnessing the Σ_2-formulas (A_n). The dialectica interpretation calls for a procedure that takes these pointwise strategies and produces a co-operative selection function F that witnesses $\forall n A_n$. Such a procedure is provided naturally by the product of selection functions.

We now prove that more generally, EPS witnesses the functional interpretation of countable dependent choice for arbitrary formulas. This is a slight reformulation of the analogous result for Spector's bar recursion first given in [61]. Dependent choice is usually given in the following form:

$$\mathsf{DC}^\rho\ :\ \forall n, x^\rho \exists y^\rho A_n(x,y) \to \forall x_0 \exists f^{\mathbb{N}\to\rho}(f(0) = x_0 \wedge \forall n A_n(f(n), f(n+1))).$$

It turns out that EPS naturally interprets a variant of DC given by

$$\mathsf{DC}^\rho_{\mathsf{seq}}\ :\ \forall s^{\rho^*} \exists x^\rho A_s(x) \to \exists f^{\mathbb{N}\to\rho} \forall n A_{[f](n)}(fn).$$

This sequential formulation of dependent choice is similar to the one considered in [82] and will be important in Chapter 8.

Lemma 5.11. DC *is equivalent to* $\mathsf{DC_{seq}}$.

Proof. First we show that $\mathsf{DC_{seq}} \Rightarrow \mathsf{DC}$. Define $B_s(y) :\equiv A_{|s|}(s_{|s|-1}, y)$ for $|s| > 0$ and $B_{\langle\rangle}(y) :\equiv A_0(x_0, y)$. Then we have $\forall n, x^\rho \exists y A_n(x, y) \to \forall s \exists y B_s(y)$ and therefore by $\mathsf{DC_{seq}}$ there exists $f^{\mathbb{N}\to\rho}$ satisfying

$$A_0(x_0, f(0)) \wedge \forall n > 0 A_n(f(n-1), fn).$$

Defining

$$\tilde{f}(n) := \begin{cases} x_0 & \text{if } n = 0 \\ f(n-1) & \text{if } n > 0 \end{cases}$$

we clearly have $\forall n A_n(\tilde{f}(n), \tilde{f}(n+1))$.

For $\mathsf{DC} \Rightarrow \mathsf{DC_{seq}}$, define

$$B_n(s^{\rho^*}, t^{\rho^*}) :\equiv (|s| = n \to |t| = n+1 \wedge s \prec t \wedge A_s(tn)).$$

Then $A_s(x) \to B_{|s|}(s, s * x) \to B_n(s, s * x)$ and therefore

$$\forall s \exists x A_s(x) \to \forall n, s \exists t B_n(s, t).$$

By DC for $s_0 = \langle\rangle$

$$\forall s \exists x A_s(x) \to \exists f^{\mathbb{N}\to\rho^*}(f(0) = \langle\rangle \wedge \forall n B_n(f(n), f(n+1)))$$

where

$$B_n(f(n), f(n+1)) \equiv |f(n)| = n \to |f(n+1)| = n+1 \wedge fn \prec f(n+1) \wedge A_{f(n)}(f(n+1)(n)).$$

By induction $|f(n)| = n$ for all n, and therefore

$$\forall s \exists x A_s(x) \to \exists f(f(0) = \langle\rangle \wedge \forall n(|f(n+1)| = n+1 \wedge fn \prec f(n+1) \wedge A_{f(n)}(f(n+1)(n))).$$

Define the function $\tilde{f}\colon \mathbb{N} \to \rho$ by $\tilde{f}(n) :\overset{\rho}{=} f(n+1)(n)$. Now, clearly $[\tilde{f}](0) = \langle\rangle = f(0)$ and because for arbitrary n we have $f(n) = [f(n+1)](n)$ it follows that

$$[\tilde{f}](n) = f(n) \to [\tilde{f}](n) = [f(n+1)](n) \to [\tilde{f}](n+1) = f(n+1)$$

and therefore by induction $[\tilde{f}](n) = f(n)$ for all n. Therefore we obtain $\forall n A_{[\tilde{f}](n)}(\tilde{f}(n))$. □

Remark 5.12. While it is clear that the direction $\mathsf{DC_{seq}} \Rightarrow \mathsf{DC}$ can be formalised in $\mathsf{WE\text{-}PA}^\omega$, as it stands our proof of $\mathsf{DC} \Rightarrow \mathsf{DC_{seq}}$ makes use of the extensionality axiom to derive

$$[\tilde{f}](n) = f(n) \wedge A_{f(n)}(f(n+1)(n))) \to A_{[\tilde{f}](n)}(\tilde{f}(n)).$$

While we don't doubt that this need for extensionality can be eliminated with a little effort, it does not concern us too much here as we are able to give a direct computational interpretation to $\mathsf{DC_{seq}}$ using EPS.

Theorem 5.13. *(a) The functional interpretation of the principle of dependent choice*

$$\mathsf{DC}_{\mathsf{seq}} \ : \ \forall s^{\rho^*} \exists x^{\rho} A_s(x) \to \exists f^{\mathbb{N} \to \rho} \forall n A_{[f](n)}(fn)$$

is realized by the dependent product of selection functions EPS*.*

(b) The functional interpretation of the principle of countable choice

$$\mathsf{AC}^{\mathbb{N}} \ : \ \forall n \exists x^{\rho} A_n(x) \to \exists f^{\mathbb{N} \to \rho} \forall n A_n(f(n))$$

is realized by the simple product of selection functions.

(c) The functional interpretation of the principle of finite choice

$$\mathsf{FAC} \ : \ \forall k (\forall n \leq k \exists x^{\rho} A_n(x) \to \exists s^{\rho^*} \forall n \leq k A_n(s_n))$$

is realized by a finite product of selection functions.

Proof. We prove (a). Part (b) was first shown in [30], and in any rate follows directly from (a). Part (c) will be discussed in Chapter 10.

Since $A_s(x)$ is interpreted as $\exists \tilde{x}^{\tilde{\rho}} \forall y^{\tau} |A_s(x)^{\mathrm{N}}|^{\tilde{x}}_{y}$ it suffices to interpret instances of $\mathsf{DC}_{\mathsf{seq}}$ for Σ_2-formulas:

$$\forall s^{\rho^*} \exists x^{\rho}, \tilde{x}^{\tilde{\rho}} \forall y^{\tau} A_s(x, \tilde{x}, y) \to \exists f^{\mathbb{N} \to \rho} \forall n \exists \tilde{x} \forall y A_{[f](n)}(fn, \tilde{x}, y).$$

Moreover, by adding a dummy variable t of type $\tilde{\rho}^*$ this in turn follows directly from an instance of Π_1-$\mathsf{DC}_{\mathsf{seq}}$:

$$\Pi_1\text{-}\mathsf{DC}_{\mathsf{seq}} \ : \ \forall s^{\rho^*}, t^{\tilde{\rho}^*} \exists x, \tilde{x} \forall y A_{s,t}(x, \tilde{x}, y) \to \exists f^{\mathbb{N} \to \rho}, \tilde{f}^{\mathbb{N} \to \tilde{\rho}} \forall n, y A_{[f, \tilde{f}](n)}(fn, \tilde{f}n, y).$$

Therefore it suffices to deal with Π_1-$\mathsf{DC}_{\mathsf{seq}}$ in general i.e.

$$\Pi_1\text{-}\mathsf{DC}_{\mathsf{seq}} \ : \ \forall s^{\rho^*} \exists x^{\rho} \forall y^{\tau} A_s(x, y) \to \exists f \forall n, y A_{[f](n)}(fn, y).$$

where A_s is quantifier-free. This has negative translation equivalent to

$$\forall s \neg\neg \exists x \forall y A_s(x, y) \to \neg\neg \exists f \forall n, y A_{[f](n)}(fn, y).$$

and therefore its functional interpretation is given by

$$|\Pi_1\text{-}\mathsf{DC}^{\mathrm{N}}|^{f,p,s}_{\varepsilon,\varphi,q} \ \equiv \ A_s(\varepsilon_s p, p(\varepsilon_s p)) \to A_{[f](\varphi f)}(f(\varphi f), qf), \tag{5.7}$$

omitting, for the sake of readability, the parameters ε, φ and q on f, p and s. Let

$$\begin{aligned} f &= \mathsf{EPS}^{\varepsilon,q,\varphi}(\langle\rangle) \\ p_s &= \lambda x \, . \, q(\mathsf{EPS}^{\varepsilon,q,\varphi}(s * x)). \end{aligned}$$

By Theorem 5.7 we have that f and $p := p_{[f](\varphi f)}$ and $s := [f](\varphi f)$ satisfy $\varepsilon_s p = f(\varphi f)$ and $p(\varepsilon_s p) = qf$, and hence, clearly witness $|\Pi_1\text{-}\mathsf{DC}^{\mathrm{N}}|^{f,p,s}_{\varepsilon,\varphi,q}$. □

Theorem 5.14 (Soundness theorem for classical analysis using EPS)**.** *Let $A(a)$ be a formula in $\mathcal{L}(\mathsf{WE\text{-}PA}^\omega)$ containing only a free. Then*

$$\mathsf{WE\text{-}PA}^\omega + \mathsf{QF\text{-}AC} + \mathsf{DC}_{\mathsf{seq}} \vdash A(a) \Rightarrow \mathsf{WE\text{-}HA}^\omega + \mathsf{EPS} \vdash \forall y |A(a)^{\mathrm{N}}|^{t(a)}_{y}$$

where t is a closed term of $\mathsf{WE\text{-}HA}^\omega + \mathsf{EPS}$ which can be extracted from the proof of A.

Proof. This follows directly from Theorem 3.11 and 5.13 (a), although technically here we only prove that the conclusion can be verified in $\mathsf{E\text{-}HA}^\omega + \mathsf{EPS}$, because full extensionality is required for our proof of Lemma 5.6. However, this instance of extensionality can be reduced to the quantifier-free rule by appealing to the trick of Kohlenbach [48, Lemma 11.5], allowing us to indeed verify the conclusion in $\mathsf{WE\text{-}HA}^\omega + \mathsf{SBR}$ (in fact a quantifier-free fragment of $\mathsf{WE\text{-}HA}^\omega$ suffices) and obtain a soundness theorem for dependent choice analogous to [48, Theorem 11.9]. □

The significance of Theorem 5.13 in conjunction with Theorem 5.10 lies in the fact that our realizer for choice also computes optimal strategies in sequential games, which means that programs extracted using EPS can often be given an intuitive game-theoretic semantics. To be more precise, given a proof that is based on an instance of countable choice, we can informally identify the functional interpretation $|A^{\mathrm{N}}|$ of the theorem A with a partially defined sequential game $\mathcal{G}_A$ (whose parameters arise from the interpreted proof):

$$\begin{tikzcd}\text{Classical theorem } A \arrow[r] & \text{Partially defined game } \mathcal{G}_A\end{tikzcd}$$

and in this sense the constructive reformulation of A can be understood in terms of optimal plays in $\mathcal{G}_A$.

5.4 Alternatives to Spector's bar recursive interpretation of analysis

We have introduced a variant of Spector's bar recursion with close links to game theory, and in the following chapters we provide concrete examples of program extraction using the product of selection functions that produce realizers with a natural game semantics. However, we also address the issue of whether or not Spector's bar recursion or the closely related EPS are always best suited for extracting programs from proofs.

Spector devised bar recursion for foundational reasons - in order to extend Gödel's proof of the consistency of Peano arithmetic to classical analysis. Bar recursion is ideal for this purpose as it provides a simple and elegant solution to the functional interpretation of the double negation shift, and therefore to both countable and dependent choice.

However, its suitability as a device for extracting intuitive and readable realizers from proofs is not so firmly established. This is particularly true when a proof makes use of the choice axiom in the guise of a different principle A, where A might be arithmetical comprehension, the variants of König's lemma, or the minimal-bad-sequence argument. In this case, while we can certainly find a bar recursive realizer for A by interpreting $\mathsf{DC} \to A$, it may be better to search for a direct realizer for A that more naturally reflects the computational content of A.

The idea of constructing new realizers to the axiom of choice and its variants for the purpose of improved program extraction is not new, and has been explored in context of the realizability interpretation of analysis in e.g. [8, 10, 11, 82]. However, few alternatives for Spector's bar recursion have been proposed for the functional interpretation of analysis.

In each of the subsequent case studies, not only do we attempt to understand the bar recursive realizer extracted from our classical proof, but we consider the possibility of replacing it with a more direct and intuitive realizer. In the case of arithmetic comprehension (Chapter 6) and Σ^0_1-WKL (Chapter 7) we characterise the behaviour of our bar recursive in terms of update procedures (similar to the learning-based realizers discussed in [3]), and for minimal-bad-sequence argument (Chapter 8) we investigate the possibility of constructing a realizer for its functional interpretation via open recursion, as done by Berger in the context of modified realizability in [10].

CHAPTER

SIX

Arithmetical Comprehension

This small chapter forms a note in which we examine the computational content of the principle of arithmetical comprehension for Σ^0_1-formulas (In fact, arithmetical comprehension $\mathsf{CA}^{\mathbb{N}}_{\mathrm{ar}}$ for arbitrary formulas is derivable in $\mathsf{WE\text{-}PA}^\omega + \Sigma^0_1\text{-}\mathsf{CA}$, see [48]). As a warm up for later chapters we give the bar recursive realizer for Σ^0_1-CA in terms of EPS and outline its game semantics. Then in the main part of the chapter we analyse the behaviour of this realizer and characterise it as a fixpoint for an intuitive learning-process.

In informal set-theoretic language arithmetical comprehension for Σ^0_1-formulas is given as follows.

Arithmetical comprehension (Σ^0_1-CA). Given an arbitrary Σ^0_1-formula $B(n) :\equiv \exists k^{\mathbb{N}} B_0(n,k)$ (for B_0 quantifier-free) we can construct a set $X \subseteq \mathbb{N}$ such that $n \in X \leftrightarrow B(n)$.

We identify subsets of $\mathbb{N}$ with their characteristic functions $t_X \colon \mathbb{N} \to \mathbb{B}$. Written formally in the language of $\mathsf{WE\text{-}PA}^\omega$, Σ^0_1-CA becomes[1]

$$\exists t^{\mathbb{N}\to\mathbb{B}} \forall n (t(n) = 0 \leftrightarrow \exists k B_0(n,k)). \tag{6.1}$$

Our first observation is that we can eliminate the two way-implication, as (6.1) follows directly from the formula

$$\exists f^{\mathbb{N}\to\mathbb{N}} \forall n (\exists k B_0(n,k) \to \exists i < f(n) B_0(n,i)), \tag{6.2}$$

[1] Note that since we can construct characteristic functions for quantifier-free formulas we could alternatively write Σ^0_1-CA as the single axiom

$$\forall f^{\mathbb{N}\times\mathbb{N}\to\mathbb{N}} \exists t \forall n (t(n) = 0 \leftrightarrow \exists k (f(n,k) = 0))$$

by setting

$$t_f(n) := \begin{cases} 0 & \text{if } \exists i < f(n) B_0(n,i) \\ 1 & \text{otherwise.} \end{cases}$$

Note that, conversely, (6.2) follows from (6.1) using QF-AC. The bounded quantifier in the conclusion of (6.2) is not strictly necessary, although in this and the following chapters we sometimes prefer to work with bounded formulas because it makes the computational interpretation slightly more intuitive, and more useful in applications. Now, (6.2) is equivalent to

$$\exists f^{\mathbb{N}\to\mathbb{N}} \forall n, k (\exists i < k B_0(n,i) \to \exists i < f(n) B_0(n,i)), \tag{6.3}$$

and the ND interpretation of (6.3) is given by

$$\forall q, \varphi \exists f \forall n \leq \varphi f (\exists i < q f B_0(n,i) \to \exists i < f(n) B_0(n,i)) \tag{6.4}$$

where q and φ have type $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$. One can rephrase this interpretation of Σ^0_1-CA as follows.

Finitary arithmetical comprehension (Fin-Σ^0_1-CA). Given arbitrary functionals $q, \varphi \colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ there exists a function $f \colon \mathbb{N} \to \mathbb{N}$ such that the set $Y_f :\equiv \{n \mid \exists i < f(n) B_0(n,i)\} \subseteq X$ is an approximation to X relative to q, φ in the sense that for all $n \leq \varphi f$ we have $n \in Y$ whenever $\exists i B_0(n,i)$ is witnessed for $i < qf$.

The idea behind this equivalent formulation of Σ^0_1-CA is that $Y \subseteq X$ is an approximation to X whose 'size' and 'depth' are calibrated by the functionals φ and q respectively. The existence of such approximations for arbitrary φ, q is classically equivalent to Σ^0_1-CA. However, while the comprehension set X cannot be effectively constructed, arbitrarily good approximations Y *can* be computed using bar recursion.

6.1 A game-theoretic interpretation of Σ^0_1-CA

We now extract directly from the classical proof of Σ^0_1-CA a bar recursive realizer for Fin-Σ^0_1-CA in terms of the product of selection functions. Bar recursive interpretations of comprehension are not new (see e.g. [48, Chapter 11.3]) - our aim here is to provide one in terms of EPS that has a natural game-theoretic reading. One obtains Σ^0_1-CA in the form of (6.3) from a instance of $\mathsf{AC}^{\mathbb{N},\mathbb{N}}$ applied to

$$\forall n \exists l \forall k (\exists i < k B_0(n,i) \to \exists i < l B_0(n,i)) \tag{6.5}$$

where (6.5) is a direct consequence of the law of excluded middle sometimes referred to as the 'drinkers paradox'. The functional interpretation of (6.5) asks for a sequence of selection functions $\varepsilon \colon J_{\mathbb{N}}\mathbb{N}$ satisfying

$$\forall n, p^{\mathbb{N}\to\mathbb{N}} (\exists i < p(\varepsilon_n p) B_0(n,i) \to \exists i < \varepsilon_n p B_0(n,i)), \tag{6.6}$$

and it is clear that

$$\varepsilon_n(p) := \begin{cases} 0 & \text{if } \forall i < p(0)\neg B_0(n,i) \\ p(0) & \text{if } \exists i < p(0) B_0(n,i) \end{cases} \tag{6.7}$$

does the job. To see this, note that if $\forall i < p(0)\neg B_0(n,i)$ then the premise of (6.6) is false for $\varepsilon_n p = 0$, and otherwise the conclusion must be true for $\varepsilon_n p = p(0)$. Now, to interpret the instance of countable choice applied to (6.5) we define

$$f_{q,\varphi} := \mathsf{EPS}^{\varepsilon,q,\varphi}(\langle\rangle).$$

By Theorem 5.7 we have $\varepsilon_n p_n = f(n)$ and $qf = p_n(\varepsilon_n p_n)$ for all $n \leq \varphi f$ (for $p_n := p_{[f](n)}$ as in Theorem 5.7), therefore by (6.6) we obtain

$$\forall n \leq \varphi f(\exists i < qfB_0(n,i) \to \exists i < f(n)B_0(n,i)), \tag{6.8}$$

and so $f_{q,\varphi}$ is a realizer for Fin-Σ^0_1-CA. We can attach the game-semantics of EPS to this particular case, where Fin-Σ^0_1-CA corresponds to a partially defined game $\mathcal{G}[q,\varphi]$ which can be described in mathematical terms as follows.

(a) The move at round n is a natural number k which in this context should be viewed as a potential bound for a witness of $\exists i B_0(n,i)$.

(b) An arbitrary play α determines a subset $\{n \mid \exists i < \alpha(n)B_0(n,i)\}$ of X.

(c) The strategy for constructing an approximation to X is given by the selection functions. At position s with $|s| = n$, provided $\varphi(\hat{s}) \geq n$ the selection function $\varepsilon_{|s|}$ defines a policy for whether or not to include n in the approximation. By default it selects 0 and attempts to get away with omitting n from the approximation. However, if the outcome $q(s * 0 * \alpha)$ (where α is the strategic extension of $s * 0$ in the sense of Section 5.2) bounds a witness for $\exists i B_0(n,i)$, ε_n accepts it must change its mind and includes n in the approximation, stealing this bound as justification.

$\mathcal{G}[q,\varphi]$	Finitary arithmetical comprehension
ε_n	decide whether to include n in approximation
EPS^ε	construct approximation to X
φ, q	determine size and depth of approximation respectively

Table 6.1: Functional interpretation of Σ^0_1-CA

Under this correspondence between plays of $\mathcal{G}[q,\varphi]$ and subsets of X, we see that an optimal play of $\mathcal{G}[q,\varphi]$ given by $f = \mathsf{EPS}^{\varepsilon,q,\varphi}(\langle\rangle)$ is an approximation to X of depth and size given by q and φ. This fact is clear in informal game theoretic terms: that fn is an optimal move at round n implies that $fn = 0$ only if $q([f](n) * 0 * \alpha_{[f](n)*0}) = qf$ does not bound a witness for $\exists i B_0(n, i)$, otherwise $fn > 0$ and must by definition bound a witness for $\exists i B_0(n, i)$.

We have taken a non-constructive derivation of Σ^0_1-CA and have quickly and easily transformed it into a constructive proof of the functional interpretation of Σ^0_1-CA that can be clearly understood in mathematical terms by using the semantics of EPS.

6.2 Fin-Σ^0_1-CA as the fixpoint of an update procedure

While EPS gives us an intuitive constructive proof of Fin-Σ^0_1-CA, it is natural to ask whether we can understand the actual *computation* that it carries out in more detail. How does the specific instance EPS^ε with ε defined as in (6.7) behave? Can we give a more illuminating characterisation of how our realizer computes approximations to Σ^0_1-CA? In this section we show that the bar recursive realizer $\mathsf{EPS}^\varepsilon(\langle\rangle)$ is exactly the fixed point of a learning process that keeps track of when the selection functions 'change their mind' and discover non-trivial constructive information.

An informal description of the realizer

To begin with, let us give a rough simulation of how our realizer constructs an approximation to Fin-Σ^0_1-CA. Given q and φ it computes the sequence $f = \mathsf{EPS}^{\varepsilon,q,\varphi}(\langle\rangle)$, a procedure in which each selection function ε_n picks a default value 0, and waits for the ε_m for $m > n$ to compute the strategic extension for this choice before it decides whether or not to change its mind. The non-trivial steps in the procedure are precisely the points at which the selection functions change their mind, as only in this case do they discover genuine constructive information.

Initially, each selection function will test 0 in turn until we reach the first point N_0 at which Spector's condition holds i.e. $\varphi(\overline{0_{\mathbb{N}\to\mathbb{N}}, N_0}) = \varphi(0) < N_0$. Clearly this point is given by $N_0 = \varphi(0) + 1$. EPS will then backtrack until it finds a point $n_0 < N_0$ such that $\exists i < q(0) B_0(n_0, i)$ and ε_{n_0} is forced to change its mind. We have two possibilities:

(1) no such n_0 exists, in which case $f = \mathsf{EPS}(\langle\rangle) = 0_{\mathbb{N}\to\mathbb{N}}$;

(2) $\varepsilon_{n_0}(p_{[0](n_0)}) = q(0)$ (for p_s as defined in Theorem 5.7) where $n_0 \leq \varphi(0)$ is the greatest such that $\exists i < q(0_{\mathbb{N}\to\mathbb{N}}) B_0(n_0, i)$.

In the case (2) let us define

$$u_1 := [0_{\mathbb{N}\to\mathbb{N}}](n_0) * \ q(0_{\mathbb{N}\to\mathbb{N}}).$$

At this point in the calculation of f the ε_n are testing the default value 0 for all $n < n_0$, and it has been decided that $next([0](n_0)) = q(0)$, or in other words $\mathsf{EPS}([0](n_0)) = \mathsf{EPS}(u_1)$.

The computation continues by calculating the strategic extension of u_1. We find the first point $N_1 > n_0$ such that $\varphi(\widehat{[\widehat{u_1}](N_1)}) = \varphi(\widehat{u_1}) < N_1$. This time, either $\varphi(\widehat{u_1}) \leq n_0$ in which case $N_1 = n_0 + 1$, or $\varphi(\widehat{u_1}) > n_0$ and $N_1 = \varphi(\widehat{u_1}) + 1$.

Putting these cases together we get $N_1 = \max\{n_0 + 1, \varphi(\widehat{u_1}) + 1\}$. As before, EPS backtracks until it finds a point $n_0 < n_1 < N_1$ such that $\exists i < q(\widehat{u_1}) B_0(n_1, i)$. If no such point exists we conclude that the strategic extension of u_1 is $0_{\mathbb{N}\to\mathbb{N}}$ i.e. $\mathsf{EPS}(u_1) = \widehat{u_1}$. The backtracking then continues until a point $n_1 < n_0$ is found satisfying $\exists i < q(\widehat{u_1}) B_0(n_1, i)$. Once more, overall there are two possibilities, according to whether or not a selection function ends up changing its mind:

(1) no such n_1 exists, in which case $f = \mathsf{EPS}(\widehat{u_1}) = \widehat{u_1}$;

(2) $\varepsilon_{n_1}(p_{[\widehat{u_1}](n_1)}) = q(\widehat{u_1})$ where $n_1 \leq \max\{n_0, \varphi(\widehat{u_1})\} = \max\{|u_1| - 1, \varphi(\widehat{u_1})\}$ is the greatest such that $n_1 \neq n_0$ and $\exists i < q(\widehat{u_1}) B_0(n_1, i)$.

In the case (2) let us define

$$u_2 := [\widehat{u_1}](n_1) * q(\widehat{u_1}).$$

We have reached the next non-trivial stage in the calculation of f, in which the selection function at n_1 has changed its mind and we have $\mathit{next}([\widehat{u_1}](n_1)) = q(\widehat{u_1})$. Now the ε_n are testing 0 for all $n < n_1$ *except* for the point $n = n_0$ if $n_0 < n_1$, as this will have already been set to the value $q(0)$. If $n_1 < n_0$, then we are now calculating the optimal extension of a new position $[0](n_1) * q(\widehat{u_1})$ and we *forget* the value that was previously assigned to n_0.

Either way we continue by calculating the strategic extension of u_2, and the algorithm follows the same pattern as before, backtracking from the point $N_2 = \max\{n_1 + 1, \varphi(\widehat{u_2}) + 1\}$ and checking whether or not the default value works at points $n < N_2$ that are being tested (i.e. $n \neq n_0, n_1$ if $n_0 < n_1$, else $n \neq n_1$). Our claim is that the process continues in this way, yielding a sequence of updates

$$u_0 = \langle\rangle \mapsto u_1 \mapsto u_2 \mapsto \ldots \mapsto u_m$$

where

$$u_{j+1} = [u_j](n_j) * q(\widehat{u_j})$$

for $n_j \leq \max\{|u_j| - 1, \varphi(\widehat{u_j})\}$ the greatest such that $u_j(n_j) = 0 \wedge \exists i < q(\widehat{u_j}) B_0(n_j, i)$ (the condition $u_j(n) = 0$ indicating that ε_n is currently in the testing stage). This corresponds to case (2) above. If no such n_j exists we have $u_{j+1} = u_j$, and this corresponds to case (1) above. We claim that $f = \widehat{u_m}$ where u_m is the fixpoint of this process.

The idea behind looking at the computation of $\mathsf{EPS}(\langle\rangle)$ in this way is that our bar recursive realizer for Fin-Σ_1^0-CA can be characterised as a learning process in the sense of [1, 2], the ‘learning’ in this case corresponding to a selection function changing its mind. We now make this characterisation of our realizer precise.

EPS as a learning process

Given a finite sequence u define $\mathrm{dom}(u) := \{i < |u| \mid u(i) > 0\}$ (note that we consider the value 0 to be 'undefined'). Now define an operation $\star : \mathbb{N}^* \times ((\mathbb{N} \times \mathbb{N}) + \{\emptyset\}) \to \mathbb{N}^*$ by[2]

$$u \star \langle n, x \rangle := [\widehat{u}](n) * x$$

and $u \star \emptyset = u$.

Definition 6.1. (a) An *update procedure* is a functional $\eta \colon \mathbb{N}^* \to (\mathbb{N} \times \mathbb{N} + \{\emptyset\})$ satisfying

$$\forall u, n, x (\eta(u) = \langle n, x \rangle \to n \notin \mathrm{dom}(u) \wedge x > 0).$$

(b) Given an update procedure η and a finite sequence of numbers s, the $\star$-*learning process* $U^{\eta,s} \colon (\rho^*)^{\mathbb{N}}$ generated by η and s is the sequence defined by

$$U^{\eta,s}_0 := s \qquad U^{\eta,s}_{i+1} := U^{\eta,s}_i \star \eta(U^{\eta,s}_i).$$

(c) The learning process $U^{\eta,s}$ has a *fixpoint* if there exists some j such that $U^{\eta,s}_{j+1} = U^{\eta,s}_j$. We write $\mathsf{fix}(\eta, s) = U^{\eta,s}_j$.

Remark 6.2. Note that while our definition of an update procedure in similar to e.g. Avigad's [4] and others used in the literature, our definition of the learning process generated by an update procedure is slightly different due to the fact the operation $\star$ performs a 'cut-off' update as opposed to an update $\oplus$ in the usual sense. We elaborate on this below.

Remark 6.3. Fixpoints of learning processes are unique, since if U_j is a fixpoint then $U_l = U_j$ for all $l > j$. Also U_j is a fixpoint iff $\eta(U_j) = \emptyset$. One direction is obvious, for the other note that if $\eta(U_j) \neq \emptyset$ then $\eta(U_j) = \langle n, x \rangle$ with $n \notin \mathrm{dom}(U_j)$ and $x > 0$ which implies that $U_{j+1}(n) = x$ and therefore $n \in \mathrm{dom}(U_{j+1})$.

Theorem 6.4. *Suppose that* $\varepsilon_n \colon J_{\mathbb{N}}\mathbb{N}$ *is defined as*

$$\varepsilon_n(p) := \begin{cases} 0 & \text{if } \neg C_n(p(0)) \\ p(0) & \text{otherwise} \end{cases}$$

where C_n *is a decidable predicate satisfying* $\forall n \neg C_n(0)$. *Define the update procedure* $\eta_{q,\varphi}$ *by*

$$\eta_{q,\varphi}(u) := \langle n, q(\widehat{u}) \rangle \text{ for } \tilde{\mu} n \leq \max\{|u| - 1, \varphi(\widehat{u})\}(n \notin \mathrm{dom}(u) \wedge C_n(q(\widehat{u}))), \text{ else } \emptyset$$

where $\tilde{\mu} n \leq m Q(n)$ *is the bounded search operator returning the greatest element satisfying* $Q(n)$ *if it exists. Note that the condition on* $C_n(0)$ *ensures that* $\eta_{q,\varphi}$ *is indeed an update*

[2] Here we casually extend our collection of finite types to include coproducts + and a null object $\emptyset$

procedure. Then, provably in $\mathsf{E\text{-}HA}^\omega + \mathsf{EPS} + \mathsf{BI}$*, the learning process* $U^{\eta_{q,\varphi},\langle\rangle}$ *has a fixpoint and*

$$\mathsf{fix}(\widehat{\eta_{q,\varphi},\langle\rangle}) = \mathsf{EPS}^{\varepsilon,q,\varphi}(\langle\rangle).$$

Proof. For given q, φ we use bar induction on the statement $P(s)$ that the learning process $U^{\eta_{|s|},s}$ has a fixpoint with $\mathsf{fix}(\widehat{\eta_{|s|},s}) = \mathsf{EPS}^{\varepsilon,q,\varphi}(s)$, for

$$\eta_m(u) := \langle n, q(\widehat{u})\rangle \text{ for } \tilde{\mu} n \in [m, \max\{|u|-1, \varphi(\widehat{u})\}](n \notin \mathrm{dom}(u) \wedge C_n(q(\widehat{u}))), \text{ else } \emptyset.$$

Clearly the theorem follows from $P(\langle\rangle)$. First we must establish that for any $\alpha\colon \mathbb{N}^{\mathbb{N}}$ there exists N with $P([\alpha](N))$. Now, provably in $\mathsf{E\text{-}HA}^\omega + \mathsf{EPS}$ there exists an N satisfying Spector's condition $\varphi(\overline{\alpha,N}) < N$ (this fact, usually given in terms of GBR, is proved for EPS in Proposition 11.1). In this case $\max\{N-1, \varphi(\overline{\alpha,N})\} < N$ and therefore $\eta_N([\alpha](N)) = \emptyset$, so $U_0^{\eta_N,[\alpha](N)} = [\alpha](N)$ is a fixpoint of $U^{\eta_N,[\alpha](N)}$, and

$$\mathsf{EPS}([\alpha](N)) = [\alpha](N) \mathbin{@} 0 = \mathsf{fix}(\widehat{\eta_N,[\alpha]}(N)).$$

Now given s assume that $P(s*x)$ holds for all x. We must prove $P(s)$. If $\varphi(\widehat{s}) < |s|$ then $P(s)$ holds by a similar argument to the one above, so we assume that $\varphi(\widehat{s}) \geq |s|$, in which case by definition of EPS:

$$\mathsf{EPS}(s) = \begin{cases} \mathsf{EPS}(s*0) & \text{if } \neg C_{|s|}(q(\mathsf{EPS}(s*0))) \\ \mathsf{EPS}(s*q(\mathsf{EPS}(s*0))) & \text{otherwise.} \end{cases}$$

Let us define $u_i := U_i^{\eta_{|s|},s}$ and $v_i := U_i^{\eta_{|s|+1},s*0}$. By the hypothesis $P(s*0)$, (v_i) has a fixpoint: let j be the least with $v_j = v_{j+1}$. Now, we claim that for all $i \leq j$:

(i) $\widehat{u_i} = \widehat{v_i}$ and therefore $\mathrm{dom}(u_i) = \mathrm{dom}(v_i)$;

(ii) $s*0 \prec \widehat{u_i}$ so in particular $|s| \notin \mathrm{dom}(u_i), \mathrm{dom}(v_i)$;

(iii) $|s| \leq \max\{|u_i|-1, \varphi(\widehat{u_i})\}$;

(iv)
$$\begin{aligned} & n \in [|s|+1, \max\{|u_i|-1, \varphi(\widehat{u_i})\}] \wedge n \notin \mathrm{dom}(u_i) \wedge C_n(q(\widehat{u_i})) \\ \leftrightarrow\ & n \in [|s|+1, \max\{|v_i|-1, \varphi(\widehat{v_i})\}] \wedge n \notin \mathrm{dom}(v_i) \wedge C_n(q(\widehat{v_i})). \end{aligned}$$

We use induction. Both (i) and (ii) are clearly true if $i = 0$ because $u_0 = s$ and $v_0 = s*0$. Furthermore, $\max\{|u_0|-1, \varphi(\widehat{u_0})\} = \max\{|v_0|-1, \varphi(\widehat{v_0})\} \geq |s|$ by our assumption that $\varphi(\widehat{s}) \geq |s|$, from which (iii) and (iv) follow.

Assume the induction hypothesis for $i < j$, and note that $\eta_{|s|+1}(v_i) \neq \emptyset$ by minimality of j. Now, using the fact that

$$(*)\quad \tilde{\mu} n \in [|s|, K] Q(n) = \begin{cases} \tilde{\mu} n \in [|s|+1, K] Q(n) & \text{if it exists} \\ |s| & \text{if } |s| \leq K \wedge Q(|s|) \\ \emptyset & \text{otherwise} \end{cases}$$

we must have

$$\eta_{|s|+1}(v_i) = \langle n, q(\widehat{v_i})\rangle \overset{(i)}{=} \langle n, q(\widehat{u_i})\rangle \overset{(iv),(*)}{=} \eta_{|s|}(u_i).$$

Therefore $u_{i+1} = [\widehat{u_i}](n) * q(\widehat{u_i}) = [\widehat{v_i}](n) * q(\widehat{v_i}) = v_{i+1}$, so (i) and (iv) clearly hold for $i+1$, by the fact that $n \geq |s|+1$ (ii) and (iii) follow.

Now, for $i = j$ we have $\eta_{|s|+1}(v_j) = \emptyset$, and by $(*)$, using (iv) along with (ii) ($|s| \notin \mathrm{dom}(u_j)$) and (iii) we obtain

$$\eta_{|s|}(u_j) = \langle |s|, q(\widehat{u_j})\rangle \text{ if } C_{|s|}(q(\widehat{u_j})) \text{ else } \emptyset.$$

There are two possible scenarios:

Case (a): $\neg C_{|s|}(q(\mathsf{EPS}(s*0)))$, which by the bar induction hypothesis

$$\widehat{u_j} \overset{(ii)}{=} \widehat{v_j} \overset{P(s*0)}{=} \mathsf{EPS}(s*0)$$

is equivalent to $\neg C_{|s|}(q(\widehat{u_j}))$. This implies that $\eta_{|s|}(u_j) = \emptyset$ and so $U^{\eta_{|s|},s}$ has fixpoint u_j and

$$\widehat{u_j} = \mathsf{EPS}(s*0) = \mathsf{EPS}(s).$$

Case (b): $C_{|s|}(q(\mathsf{EPS}(s*0)))$ which as before is equivalent to $C_{|s|}(q(\widehat{u_j}))$. This time $\eta_{|s|}(u_j) = \langle |s|, q(\widehat{u_j})\rangle$, so

$$u_{j+1} = [\widehat{u_j}](|s|) * q(\widehat{u_j}) \overset{(ii)}{=} s * q(\widehat{u_j}).$$

Define $u'_i := u_{j+1+i}$ and $w_i := U^{\eta_{|s|+1}, s*q(\widehat{u_j})}$. By the hypothesis $P(s*q(\widehat{u_j}))$ we can assume that w_i has a fixpoint, so let k be the least with $w_k = w_{k+1}$. Now we use an analogous but simpler argument to before, where we can claim that for all $i \leq k$ that:

(i) $u'_i = w_i$ and therefore clearly

$$\begin{aligned} & n \in [|s|+1, \max\{|u'_i| - 1, \varphi(\widehat{u'_i})\}] \wedge n \notin \mathrm{dom}(u'_i) \wedge C_n(q(\widehat{u'_i})) \\ \leftrightarrow\ & n \in [|s|+1, \max\{|w_i| - 1, \varphi(\widehat{w_i})\}] \wedge n \notin \mathrm{dom}(w_i) \wedge C_n(q(\widehat{w_i})). \end{aligned}$$

(ii) $|s| \in \mathrm{dom}(u_i), \mathrm{dom}(w_i)$;

These are true for $i = 0$ since $u'_0 = s * q(\widehat{u_j}) = w_0$. Assuming they hold for $i < k$, by minimality of k we have

$$\eta_{|s|+1}(w_i) = \langle n, q(\widehat{w_i})\rangle \overset{(i)}{=} \langle n, q(\widehat{u'_i})\rangle \overset{(i),(*)}{=} \eta_{|s|}(u'_i)$$

and therefore $u'_{i+1} = [\widehat{u'_i}](n) * q(\widehat{u'_i}) = [\widehat{w_i}](n) * q(\widehat{w_i}) = w_{i+1}$ and (ii) follows because $n \geq |s|+1$.

For $i = k$ we have $\eta_{|s|+1}(w_k) = \emptyset$, and by $(*)$, using (iv) along with the fact that this time (ii) ($|s| \in \mathrm{dom}(u'_k)$) we have $\eta_{|s|}(u'_k) = \eta_{|s|}(u_{j+1+k}) = \emptyset$ and therefore $U^{\eta_{|s|},s}$ has fixpoint u_{j+1+k} and

$$\widehat{u_{j+1+k}} \overset{(i)}{=} \widehat{w_k} \overset{P(s*q(\widehat{u_j}))}{=} \mathsf{EPS}(s * q(\widehat{u_j})) \overset{P(s*0)}{=} \mathsf{EPS}(s * q(\mathsf{EPS}(s * 0))) = \mathsf{EPS}(s).$$

This completes the bar induction, and hence the proof. □

Setting $C_n(x) := \exists i < x B_0(n, i)$, Theorem 6.4 gives us an alternative characterisation of our bar recursive realizer of $\mathsf{Fin}\text{-}\Sigma^0_1\text{-}\mathsf{CA}$. The functional interpretation of $\Sigma^0_1\text{-}\mathsf{CA}$ asks for a finite approximation $Y_{q,\varphi}$ to the non-computable set $X := \{n \mid B(n)\}$ on arbitrary counterexample functions q, φ. The bar recursive realizer extracted to produce Y returns the fixpoint of a learning process $U^{\eta,\langle\rangle}$, which in set theoretic terms can be written as

$$\emptyset \mapsto Y_1 \mapsto Y_2 \mapsto \ldots \mapsto Y_m = Y.$$

where $Y_i := \{n \mid \exists i < \widehat{U_i^{\eta,\langle\rangle}}(n) B_0(n, i)\}$. This learning process starts with the empty set and by a pattern of trial and correction picks up new elements until the process terminates. It is clear from this description why our realizer works. We can informally write $Y_{i+1} = [Y_i](n_i) \cup \{n_i\}$ for $n_i \in X$, therefore by induction Y_i is a subset of the non-computable set X for all i. In addition, $\eta(U_m) = \emptyset$ implies that, setting $f = \widehat{U_m}$,

$$\forall n \leq \varphi f (n \notin Y_m \to \forall i < q f \neg B_0(n, i)),$$

which is precisely the condition needed for Y_m to be an approximation to X.

We note that inessential adjustments to the bar recursive realizer - for instance altering the selection functions so that they play a default value 1 instead of 0, might lead to a realizer that is slightly more difficult to describe in these terms, but would ultimately behave in a broadly similar fashion.

Viewing the bar recursive interpretation of $\Sigma^0_1\text{-}\mathsf{CA}$ as the fixpoint of a learning process not only improves our understanding of how programs from proofs involving $\Sigma^0_1\text{-}\mathsf{CA}$ behave, but brings the ND interpretation and bar recursion closer to the world of learning-based realizability that has been extensively explored in one sense or another in [2, 3, 4, 22] to name just a few. It is not surprising in itself that Spector's bar recursion can be characterised in terms of learning - indeed it has been suggested that the concept of learning underlies and unites all computational interpretations of classical logic (see e.g. [2]). Nevertheless it is illuminating to be able to describe in more concrete terms the variation on learning that specific computational interpretations implement, as we have done in Theorem 6.4.

We will not go into any detail here on how the combination of negative translation and functional interpretation compare with the more direct learning-based realizability interpretations of classical logic that have been produced in the last decade or so. We point out that the learning process defined in this section is based on an operation $\star$ which only retains constructive information *below* the point at which we're updating. This 'forgetful' behaviour is an intrinsic feature of the backtracking implemented by bar recursion. In contrast, learning in the sense of [3, 4] is based on an operation $\oplus$ is given by

$$(\alpha \oplus \langle n, x\rangle)(m) := \begin{cases} x & \text{if } m = n \\ \alpha(n) & \text{otherwise.} \end{cases}$$

where α is a partial function of type $\mathbb{N} \to \mathbb{N}$. The operator $\oplus$ generates learning processes that always retain any constructive information they find.

All this raises an interesting question of whether the functional interpretation is of Σ^0_1-CA is better witnessed directly using a different form of recursion based on learning processes, not only to produce more expressive and intuitive realizers, but for the sake of efficiency too. However, we leave a more detailed study of this rather subtle issue to future work.

CHAPTER

SEVEN

Weak König's Lemma for Undecidable Trees

In the previous chapter we gave a bar recursive computational interpretation of arithmetical comprehension that could be understood in intuitive game-theoretic terms. In this chapter we examine, from the same perspective, the computational content of weak König's lemma for Σ^0_1-trees. We give some applications in mathematics, in each case providing an informal game-theoretic description of the extracted realizers.

When we talk of weak König's lemma in the context of reverse mathematics we usually mean weak König's lemma restricted to quantifier-free trees (as in Section 2.3). Weak König's lemma for undecidable Σ^0_1-trees (Σ^0_1-WKL) is considerably stronger and can be used to conveniently formalise several important theorems of mathematics including the Bolzano-Weierstrass theorem and the infinite Ramsey's theorem, as has been done in [78] and [57] respectively - bar recursive realizers for these principles also being extracted in [78] and [56] respectively.

In this Chapter we focus on investigating the semantics of the general bar recursive interpretation of Σ^0_1-WKL, and as applications we aim to derive readable constructive proofs of the finitizations of these two theorems given by the functional interpretation.

Weak König's lemma for Σ^0_1 trees (Σ^0_1-WKL). Any infinite binary Σ^0_1-tree T has an infinite branch α.

We write $T(s^{\mathbb{B}^*}) :\equiv \exists k T_0(s,k)$ where $T_0(s,i)$ is quantifier-free. Without loss of generality we can insist on monotonicity in the second argument by setting $T_0'(s,k) :\equiv \exists i < k T_0(s,i)$, which will turn out to be quite convenient when extracting our realizer. For Σ^0_1-trees, prefix closure $\forall s,t(T(s*t) \to T(s))$ is no longer a purely universal statement, and our realizer is constructed relative to a functional $\theta\colon \mathbb{B}^* \times \mathbb{N} \to \mathbb{N}$ that witnesses closure of T in the following sense:

$$\forall s,t,k(\exists i < k T_0(s*t,i) \to \exists i < \theta(s*t,k) T_0(s,i)). \tag{7.1}$$

We can write Σ^0_1-WKL formally as

$$\Sigma^0_1\text{-}\mathsf{WKL}\ :\ \forall n \exists s \in \mathbb{B}^n, kT_0'(s,k) \to \exists \alpha^{\mathbb{B}^{\mathbb{N}}}, \beta^{\mathbb{N}^{\mathbb{N}}} \forall n T_0'([\alpha](n), \beta(n)).$$

where in fact the conclusion is slightly strengthened through an application of QF-AC to obtain the function β. As with WKL we can absorb the bounded quantifier $\exists s \in \mathbb{B}^n$ by defining

$$h_k(n) := \mu s \in \mathbb{B}^n \exists i < k T_0(s,i) \text{ else } 0$$

so that Σ^0_1-WKL becomes

$$\Sigma^0_1\text{-}\mathsf{WKL}\ :\ \forall n \exists k T_0'(h_k n, k) \to \exists \alpha, \beta \forall n T_0'([\alpha](n), \beta(n)).$$

Remark 7.1. As with Σ^0_1-CA we could alternatively write Σ^0_1-WKL as a single axiom i.e.

$$\begin{cases} \forall f[\forall s,t(\exists k(f(s*t,k)=0) \to \exists k(f(s,k)=0)) \\ \to (\forall n \exists s,k(f(s,k)=0) \to \exists \alpha \forall n \exists k(f([\alpha](n),k)=0))] \end{cases}$$

This is (essentially) the form in which it is interpreted in e.g. [78]. The difference here is that we interpret a *rule* form of Σ^0_1-WKL in which we assume that the predicate T is prefix closed. This is certainly sufficient for applications of Σ^0_1-WKL to specific trees, and it means that we have the luxury of a somewhat simpler program extraction. Note, however, that by calibrating exactly where our realizers for the rule form use prefix closure, we could easily regain a functional interpretation of the *axiom* form of Σ^0_1-WKL.

It is easy to see that Σ^0_1-WKL can be reduced to WKL by encoding finite branches $s\colon \mathbb{B}^*$ as natural numbers and using a single instance of Σ^0_1-CA to produce a characteristic function for T. We analyse the computational content of a slightly different reduction of Σ^0_1-WKL to WKL using bounded collection and Π^0_1-AC.

Theorem 7.2. *Σ^0_1-WKL is instancewise provable from Π^0_1-AC over WE-$\mathsf{PA}^\omega + \mathsf{QF}$-$\mathsf{AC} +$ WKL.*

Proof. By the law of excluded middle we have

$$\forall n \forall s^{\mathbb{B}^n} \exists l \forall k (\exists i < k T_0(s,i) \to \exists i < l T_0(s,i)).$$

Applying bounded collection over the quantifier $\forall s \in \mathbb{B}^n$ we have

$$\forall n \exists L \forall s^{\mathbb{B}^n}, k(\exists i < k T_0(s,i) \to \exists i < L T_0(s,i))$$

and therefore by Π^0_1-AC there exists $\beta\colon \mathbb{N} \to \mathbb{N}$ satisfying

$$\forall s, k(\exists i < k T_0(s,i) \to \exists i < \beta(|s|) T_0(s,i)). \tag{7.2}$$

Observe that this is just a reformulation of (6.2) where we now have an instance of bounded collection as an intermediate step. Now, if T is infinite, then by (7.2) $T^\beta(s) :\equiv T'_0(s, \beta(|s|))$ is an infinite quantifier-free tree. By WKL on T^β there exists a $\alpha\colon \mathbb{B}^\mathbb{N}$ satisfying $\forall n T^\beta([\alpha](n))$ or in other words

$$\forall n T'_0([\alpha](n), \beta(n))$$

which is the conclusion of Σ^0_1-WKL. □

7.1 A bar recursive interpretation of Σ^0_1-WKL

Using MP^ω, the negative translation of Σ^0_1-WKL is equivalent to

$$\forall n \exists k T'_0(h_k n, k) \to \neg\neg\exists \alpha, \beta \forall n T'_0([\alpha](n), \beta(n)),$$

so we will construct a realizer for the functional interpretation of

$$\exists g^{\mathbb{N}^\mathbb{N}} \forall n T'_0(h_{gn} n, gn) \to \forall \omega \exists \alpha, \beta \forall n \leq \omega\alpha\beta T'_0([\alpha](n), \beta(n)) \tag{7.3}$$

where $\omega\colon \mathbb{B}^\mathbb{N} \times \mathbb{N}^\mathbb{N} \to \mathbb{N}$, which will entail the extraction of α, β and n on parameters g, ω and θ.

Remark 7.3. Note that technically the functional interpretation only requires us to realize

$$\exists g^{\mathbb{N}^\mathbb{N}} \forall n T'_0(h_{gn} n, gn) \to \forall \tilde{\omega} \exists \alpha, \beta T'_0([\alpha](\tilde{\omega}\alpha\beta), \beta(\tilde{\omega}\alpha\beta)). \tag{7.4}$$

However, bar recursion naturally provides us with a realizer for (7.3), and either way a realizer for (7.3) is easily obtained from a realizer for (7.4) by setting $\tilde{\omega}\alpha\beta := \mu n \leq \omega\alpha\beta \neg T'_0([\alpha](n), \beta(n))$.

One can rephrase (7.3) more naturally as follows

Finitary weak König's lemma for Σ^0_1 trees (Fin-Σ^0_1-WKL). Suppose there exists a function $g\colon \mathbb{N} \to \mathbb{N}$ witnessing that the Σ^0_1-tree T has branches of arbitrary length. Then for any $\omega\colon \mathbb{B}^\mathbb{N} \times \mathbb{N}^\mathbb{N} \to \mathbb{N}$ there exists a sequence $\alpha\colon \mathbb{B}^\mathbb{N}$ along with a function $\beta\colon \mathbb{N}^\mathbb{N}$ witnessing that α is an approximation to an infinite branch of T, in the sense that for all $n \leq \omega\alpha\beta$, $\beta(n)$ bounds a witness for $[\alpha](n) \in T$.

We now extract from the classical proof of Σ^0_1-WKL a constructive proof of (the full functional interpretation of) Fin-Σ^0_1-WKL. Our first step is the construction of an approximation to the function β in (7.2) using the product of selection functions.

Lemma 7.4. *For arbitrary $q, \varphi\colon \mathbb{N}^\mathbb{N} \to \mathbb{N}$ there exists $\beta_{q,\varphi}\colon \mathbb{N} \to \mathbb{N}$ satisfying*

$$\forall n \leq \varphi\beta \forall s \in \mathbb{B}^n (T'_0(s, q\beta) \to T'_0(s, \beta_{q,\varphi} n)).$$

Proof. Define the selection functions $\varepsilon_n \colon J_{\mathbb{N}}\mathbb{N}$ by

$$\varepsilon_n(p) := p^{(i)}(0) \tag{7.5}$$

where $i \leq 2^n$ is the least such that

$$\forall s \in \mathbb{B}^n (T_0'(s, p^{(i+1)}(0)) \to T_0'(s, p^{(i)}(0))).$$

It is easy to see that ε_n is well-defined. If not, then we end up with a sequence $s_0, \ldots, s_{2^n}$ in $\mathbb{B}^n$ such that

$$\exists i < p^{(j+1)}(0) T_0(s_j, i) \wedge \forall i < p^{(j)}(0) \neg T_0(s_j, i),$$

for all j, which implies that the s_j form $2^n + 1$ distinct elements of $\mathbb{B}^n$ - a contradiction. Therefore the ε_n satisfy

$$\forall n, p \forall s \in \mathbb{B}^n (T_0'(s, p(\varepsilon_n p)) \to T_0'(s, \varepsilon_n p))$$

Now define $\beta_{q,\varphi} := \mathsf{EPS}^{\varepsilon,q,\varphi}(\langle\rangle)$. By the fact that EPS solves Spector's equations (5.3) i.e. $\beta_{q,\varphi} n = \varepsilon_n p_n$ and $p_n(\varepsilon_n p_n) = q\beta_{q,\varphi}$ for all $n \leq \varphi\beta_{q,\varphi}$, we have

$$\forall s \in \mathbb{B}^n (T_0'(s, q\beta_{q,\varphi}) \to T_0'(s, \beta_{q,\varphi} n))$$

for all $n \leq \varphi\beta_{q,\varphi}$. □

Finally, we use Howard's computational interpretation of WKL (Section 4.3) in conjunction with our approximation of β to witness $\mathsf{Fin}\text{-}\Sigma^0_1\text{-}\mathsf{WKL}$. Informally, we assume the existence of a non-computable function β satisfying

$$\forall s, k (\exists i < k T_0(s, k) \to \exists i < \beta(|s|) T_0(s, i))$$

and compute an approximation α^β to an infinite branch in the tree $T^\beta(s) \equiv \exists i < \beta(|s|) T_0(s, i)$ on the functional $\omega_\beta := \lambda\alpha.\omega\alpha\beta$. We then calibrate the size q_ω, φ_ω of the approximation to β necessary to validate our realizer. Of course, in pure logical terms this is nothing more than the functional interpretation of implication. We (very informally) illustrate the whole construction in Figure 7.1.

Theorem 7.5. *Suppose we have a binary tree T, whose prefix closure is witnessed by θ as in (7.1). Given parameters $g \colon \mathbb{N} \to \mathbb{N}$ and $\omega \colon \mathbb{B}^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$, for arbitrary $\beta \colon \mathbb{N} \to \mathbb{N}$ define K^β_ω using Howard's binary bar recursion as*

$$K^\beta_\omega(s) := \begin{cases} 0 & \text{if } \omega(\hat{s}, \beta) < |s| \\ 1 + \max\{K^\beta_\omega(s * 0), K^\beta_\omega(s * 1)\} & \text{otherwise.} \end{cases}$$

$$\dfrac{\overset{\text{Th. 4.7}}{\lambda\beta\ .\ |\mathsf{WKL}[\beta]|^{\alpha_\omega,\beta}_{\omega_\beta}} \qquad \overset{\text{L. 7.4}}{|\Sigma^0_1\text{-}\mathsf{CA}|^{\beta}_{q_\omega,\varphi_\omega}}}{|\Sigma^0_1\text{-}\mathsf{WKL}|^{\alpha_\beta,\beta}_{\omega}}\ \text{Th. 7.5}$$

Figure 7.1: Constructive proof of finitary weak König's lemma for Σ^0_1-trees

Then define $N^\beta_\omega := K^\beta_\omega(\langle\rangle)$ *and* $\alpha^\beta_{\omega,g} := \widehat{u^\beta_{\omega,g}}$ *where* $u^\beta_{\omega,g}$ *is the least* $u \preceq h_{gN^\beta_\omega} N^\beta_\omega$ *satisfying* $\omega(\widehat{u},\beta) < |u|$ *(which must exist by the proof of Theorem 4.7). Finally, abbreviate* $\beta_{\omega,g} := \beta_{q_{\omega,g},\varphi_{\omega,g}}$ *as in Lemma 7.4 where*

$$\begin{aligned} q_{\omega,g}\beta &:= \theta(h_{gN^\beta_\omega} N^\beta_\omega, gN^\beta_\omega) \\ \varphi_{\omega,g}\beta &:= \omega\alpha^\beta_{\omega,g}\beta. \end{aligned}$$

Then $N_{\omega,g} := N^{\beta_{\omega,g}}_\omega$, $\alpha_{\omega,g} := \alpha^{\beta_{\omega,g}}_{\omega,g}$ *and* $\beta_{\omega,g}$ *witness the functional interpretation of (7.3) i.e.*

$$\forall g, \omega(T'_0(h_{gN_{\omega,g}} N_{\omega,g}, gN_{\omega,g}) \to \forall n \leq \omega\alpha_{\omega,g}\beta_{\omega,g} T'_0([\alpha_{\omega,g}](n), \beta_{\omega,g}n).$$

Proof. Let $n \leq \omega\alpha_{\omega,g}\beta_{\omega,g}$. Then because $\alpha_{\omega,g} = \widehat{u_{\omega,g}}$ where we abbreviate $u_{\omega,g} := u^{\beta_{\omega,g}}_{\omega,g}$ we have,

$$[\alpha_{\omega,g}](n) \preceq [\alpha_{\omega,g}](\omega\alpha_{\omega,g}\beta_{\omega,g}) = [\widehat{u_{\omega,g}}](\omega(\widehat{u_{\omega,g}}, \beta_{\omega,g})) \overset{\omega(\widehat{u},\beta)<|u|}{\prec} u_{\omega,g} \preceq h_{gN_{\omega,g}} N_{\omega,g}.$$

Therefore by prefix closure of T (witnessed by θ) and definition of $q_{\omega,g}$,

$$T'_0(h_{gN_{\omega,g}} N_{\omega,g}, gN_{\omega,g}) \to T'_0([\alpha_{\omega,g}](n), \theta(h_{gN_{\omega,g}} N_{\omega,g}, gN_{\omega,g})) \to T'_0([\alpha_{\omega,g}](n), q_{\omega,g}\beta_{\omega,g}).$$

Now by Lemma 7.4 and the fact that $\omega\alpha_{\omega,g}\beta_{\omega,g} = \varphi_{\omega,g}\beta_{\omega,g}$, we have

$$\forall n \leq \omega\alpha_{\omega,g}\beta_{\omega,g} \forall s \in \mathbb{B}^n (T'_0(s, q_{\omega,g}\beta_{\omega,g}) \to T'_0(s, \beta_{\omega,g}n)),$$

so setting $s = [\alpha_{\omega,g}](n)$ we get

$$T'_0([\alpha_{\omega,g}](n), q_{\omega,g}\beta_{\omega,g}) \to T'_0([\alpha_{\omega,g}](n), \beta_{\omega,g}n)$$

and putting these together we obtain

$$T'_0(h_{gN_{\omega,g}} N_{\omega,g}, gN_{\omega,g}) \to T'_0([\alpha_{\omega,g}](n), \beta_{\omega,g}n)$$

for arbitrary $n \leq \omega\alpha_{\omega,g}\beta_{\omega,g}$, and we're done. □

A game theoretic description of the realizer

We now make explicit the game semantics of our realizer for $\mathsf{Fin}\text{-}\Sigma^0_1\text{-}\mathsf{WKL}$. The situation is more complex than for arithmetical comprehension in the previous chapter, partly due to the fact that we now have two instances of bar recursion. However, the most significant component of the realizer is the single instance of EPS used in Lemma 7.4 to construct the comprehension function β, and this behaves similarly to our realizer in Chapter 6, as the selection functions (7.5) are just an iterative version of (6.7). Our constructive proof of $\mathsf{Fin}\text{-}\Sigma^0_1\text{-}\mathsf{WKL}$ can be described in terms of a central sequential game $\mathcal{G}[\omega, g] = (\varepsilon, q_{\omega,g}, \varphi_{\omega,g})$ of type $(\mathbb{N}, \mathbb{N})$ parametrised by ω and g.

(a) A play β of $\mathcal{G}$ defines a subset $T^\beta = \{s \mid \exists i < \beta(|s|) T_0(s, i)\}$ of the non-computable tree T, where moves at round n are natural numbers that witness branches of length n. Note that for arbitrary β the set T^β need not be a tree itself.

(b) Under the assumption that g satisfies the premise of (7.3), for each play β we can construct, via binary bar recursion, a branch $\alpha^\beta_{\omega,g}$ satisfying $T([\alpha^\beta_{\omega,g}](\omega\alpha^\beta_{\omega,g}\beta))$.

Now, the property '$[\alpha^\beta_{\omega,g}](\omega\alpha^\beta_{\omega,g}\beta)$ is a branch of T' is substantially weaker than $[\alpha^\beta_{\omega,g}](\omega\alpha^\beta_{\omega,g}\beta)$, β being an *approximation* to an infinite branch in the sense of $\mathsf{Fin}\text{-}\Sigma^0_1\text{-}\mathsf{WKL}$. For this, we require that β also satisfies $\forall n \leq \omega\alpha^\beta_{\omega,g}\beta T_0'([\alpha^\beta_{\omega,g}](n), \beta n)$, or in other words

$$\forall n \leq \omega\alpha^\beta_{\omega,g}\beta([\alpha^\beta_{\omega,g}](n) \in T^\beta).$$

One way of looking at β is that it allows us to compute arbitrarily many witnesses that in a sense *converge* to $\alpha^\beta_{\omega,g}$.

The role of our main instance of EPS is to compute an optimal play $\beta_{q,\varphi}$ in $\mathcal{G} = (\varepsilon, q, \varphi)$ that will form an approximation to the ideal characteristic function β in (7.2) relative to given counterexample functions q and φ. We calibrate the size of this approximation in terms of ω and g to ensure that $T^{\beta_{q,\varphi}}$ contains $[\alpha^{\beta_{q,\varphi}}_{\omega,g}](n)$ for $n \leq \omega\alpha^{\beta_{q,\varphi}}_{\omega,g}\beta_{q,\varphi}$, and therefore contains an approximation to an infinite branch.

(c) The selection functions ε defined in Lemma 7.4 build an approximation $\beta_{q,\varphi}$ relative to some φ, q using the following strategy. At position $t^{\mathbb{N}^*}$ with $|t| = n$ and $\varphi(\widehat{t}) \geq n$, the selection function ε_n decides which branches of length n to include in the approximation. It plays a default value $x_0 = 0$ and tries to get away with omitting all branches in $\mathbb{B}^n$ from the approximation. If, however, the outcome $p(0) = q(\mathsf{EPS}(t * 0)) = q(t*0*\alpha_0)$ (where α_0 is the optimal extension of $t*0$) bounds a witness for $T_0(s, i)$ for some $s \in \mathbb{B}^n$, it changes its mind and updates $x_0 \mapsto x_1 = q(t * 0 * \alpha_0)$ to include all such s in the approximation. It now repeats this step: if $p(p(0)) = p(x_1) = q(t*x_1*\alpha_1)$ (where α_1 the optimal continuation of $t * x_1$) bounds a witness for any *new* branches not already witnessed by x_1, ε_n updates $x_1 \mapsto x_2 = q(t * x_1 * \alpha_1)$ and adds those new branches to the approximation. It continues the process of trial and improvement at most 2^n times before it find some x_j bounding a suitably large subset of $\mathbb{B}^n$ such that

$$\exists i < q(t * x_j * \alpha_j)T_0(s,i) \to \exists i < x_j T_0(s,i)$$

for all $s \in \mathbb{B}^n$.

An optimal play $\beta_{q,\varphi}$ in the game satisfies

$$n \leq \varphi\beta_{q,\varphi}, \forall s \in \mathbb{B}^n[\exists i < q\beta_{q,\varphi}T_0(s,i) \to \exists i < \beta_{q,\varphi}nT_0(s,i)].$$

In other words, the set $T^{\beta_{q,\varphi}}$ is an approximation to T of size $\varphi\beta_{q,\varphi}$ and depth $q\beta_{q,\varphi}$. Now, given ω and g, specific parameters $q_{\omega,g}$, $\varphi_{\omega,g}$ for $\mathcal{G}[\omega,g]$ are given as follows.

(d) The *relevant part* of a play β, which determines the size of the approximation to T, is given by $\varphi_{\omega,g}\beta := \omega\alpha^{\beta}_{\omega,g}\beta$ to guarantee that the approximation covers the branches $[\alpha^{\beta}_{\omega,g}](n)$ for $n \leq \omega\alpha^{\beta}_{\omega,g}\beta$.

(e) The *outcome* of a play β, which determines the 'depth' of the approximation to T, is given by $q_{\omega,g}\beta := \theta(h_{gN^{\beta}_{\omega}}N^{\beta}_{\omega}, gN^{\beta}_{\omega})$, which guarantees that the approximation also covers witnesses to $[\alpha^{\beta}_{\omega,g}](n)$ for all $n \leq \omega\alpha^{\beta}_{\omega,g}\beta$, since $T'_0(h_{gN^{\beta}_{\omega}}N^{\beta}_{\omega}, gN^{\beta}_{\omega})$ implies $T'_0([\alpha^{\beta}_{\omega,g}](n), q_{\omega,g}\beta)$.

Together, these imply that an optimal play $\beta_{\omega,g} = \beta_{q_{\omega,g},\varphi_{\omega,g}}$ of $\mathcal{G}[\omega,g]$ satisfies

$$\exists i < \beta_{\omega,g}nT_0([\alpha^{\beta_{\omega,g}}_{\omega,g}](n), i)$$

for all $n \leq \omega\alpha^{\beta_{\omega,g}}_{\omega,g}\beta_{\omega,g}$ and so forms an approximation to T^{β} of sufficiently high quality that it contains an approximation to an infinite branch of T relative to ω, hence realizing the functional interpretation of Σ^0_1-WKL.

The semantics of EPS give us a clear game-theoretic characterisation of our constructive proof of Fin-Σ^0_1-WKL, and one which nicely reflects the structure of the classical proof it was extracted from. Our derivation of Σ^0_1-WKL in Theorem 7.2 contains three key ineffective steps: first a combination of the law of excluded middle and bounded collection to construct a bound L for witnesses $T(s)$ for s of fixed length, then an application of Π^0_1-AC to obtain a characteristic function β for T, and finally weak König's lemma applied to the quantifier-free tree T^{β}.

These three steps correspond to the main components of the game $\mathcal{G}[\omega,g]$. Bounded collection is interpreted by the selection functions ε_n which define a strategy for $\mathcal{G}[\omega,g]$ that entails approximately deciding which branches of length n lie in T, and their unbounded product $\mathsf{EPS}^{\varepsilon}$ interprets Π^0_1-AC to form an approximation to the characteristic function β. Finally, the outcome and control functionals $\varphi_{\omega,g}$ and $q_{\omega,g}$ constructed using binary bar recursion interpret WKL and ensure that T^{β} contains an approximation to an infinite branch of T.

classical proof	sequential game $\mathcal{G}[\omega, g]$
LEM/bounded collection	selection functions ε_n
countable choice	unbounded product $\mathsf{EPS}^{\varepsilon}$
weak König's lemma	outcome and control functionals $q_{\omega,g}$, $\varphi_{\omega,g}$

Table 7.1: Functional interpretation of Σ^0_1-WKL

Fin-Σ^0_1-WKL as the fixpoint of an update procedure

We claim that the ideas from Section 6.2, in which the realizer for Σ^0_1-CA was characterised in terms of a learning process, are also valid for our realizer of Σ^0_1-WKL whose selection functions ε employ a similar strategy. Our approximation to T could be characterised as the fixpoint of a $\star$-learning process

$$T_0 = \emptyset \mapsto T_1 \mapsto T_2 \mapsto \ldots \mapsto T_m \supseteq [\alpha](\omega\alpha\beta)$$

where T_{i+1} is obtained from T_i by adding (potentially several) new branches of length n to T_i and erasing all branches of length greater than n. In contrast to the selection functions defined in Chapter 6, our selection functions here test up to 2^n values before deciding on a move, suggesting that our update procedure would allow updated points to be 'improved' finitely many times. It would be interesting to see an extension of Theorem 6.4 that incorporates our realizer for Σ^0_1-WKL

7.2 The Bolzano-Weierstrass theorem

We now give the first of two applications in which we briefly illustrate how the game semantics of our realizer of Σ^0_1-WKL might be useful in practise to shed some light on the computational content of well-known theorems in mathematics. The Bolzano-Weierstrass principle (BW) follows directly from Σ^0_1-WKL, and we give a straightforward computational interpretation to BW for rational sequences in the unit interval. The reader is directed to [78] for a comprehensive account of the computational content of the Bolzano-Weierstrass principle. There a general bar recursive realizer for the ND interpretation of BW for sequences of reals in the product space $\Pi_{n\in\mathbb{N}}[-k_n, k_n]$ is given, and a complexity analysis of the realizer is carried out to calibrate the computational contribution of fixed instances of BW used in proofs of Π_2-theorems. However, the realizer given in [78] is rather intricate, and we feel it is beneficial to present a realizer in terms of EPS for a simple instance of BW and describe how it behaves.

Bolzano-Weierstrass principle for $[0,1]_{\mathbb{Q}}$**.** Any sequence (x_i) of rationals in the unit interval $[0,1]$ contains a subsequence (x_{bi}) converging to some $a \in [0,1]$.

We can express this formally as

$$\mathsf{BW}\ :\ \forall (x_i)^{\mathbb{N}\to[0,1]_{\mathbb{Q}}}\exists \alpha^{\mathbb{B}^{\mathbb{N}}}, b^{\mathbb{N}^{\mathbb{N}}}\forall n(bn < b(n+1) \wedge x_{bn} \in I_{[\alpha](n)}),$$

where for a finite sequence of booleans s we define the interval

$$I_s := \left[\sum_{i=0}^{|s|-1}\frac{s_i}{2^{i+1}}, \sum_{i=0}^{|s|-1}\frac{s_i}{2^{i+1}} + \frac{1}{2^{|s|}}\right]$$

for $|s| > 0$ and $I_{\langle\rangle} := [0,1]$. Note that $x \in I_s$ is decidable for $x \colon \mathbb{Q}$. Intuitively we identify $\alpha \colon \mathbb{B}^{\mathbb{N}}$ with the real number $a := \sum_{i=0}^{\infty}\frac{\alpha_i}{2^{i+1}}$, so that $x_{bn} \in I_{[\alpha](n)} \leftrightarrow |x_{bn} - a| \leq 2^{-n}$ and therefore (x_{bi}) converges to a (although a detailed formalisation of BW would require us to make this intuition explicit relative to some appropriate encoding of the real numbers - see [78] for details).

Theorem 7.6. BW *is instancewise provable from* Σ^0_1-WKL *over* $\mathsf{WE\text{-}PA}^\omega + \mathsf{QF\text{-}AC}$.

Proof. Fix the sequence (x_i) and define

$$T(s) :\equiv \exists i(\underbrace{|s| \leq i \wedge (x_i \in I_s)}_{T_0(s,i)}).$$

Observe that the functional θ witnessing prefix closure of T can be taken to be the identity in this case, as $T_0(s * t, i) \to T_0(s,i)$ for all s, t because $I_{s*t} \subseteq I_s$. The tree T is infinite because for any n there is some sequence s of length n such that the interval I_s contains the point x_n. Therefore by Σ^0_1-WKL there exists $\alpha \colon \mathbb{B}^{\mathbb{N}}$ and $\beta \colon \mathbb{N}^{\mathbb{N}}$ satisfying

$$\forall n \exists i \in [n, \beta(n))(x_i \in I_{[\alpha](n)}).$$

Now define $b_{\alpha,\beta} \colon \mathbb{N} \to \mathbb{N}$ primitive recursively by

$$\begin{aligned} b_{\alpha,\beta}(0) &:= 0 \\ b_{\alpha,\beta}(n+1) &:= \mu i \in [b(n)+1, \beta(b(n)+1))(x_i \in I_{[\alpha](b(n)+1)}). \end{aligned}$$

Then $b(n) < b(n+1)$ by definition. Moreover, $x_{b(0)} \in I_{\langle\rangle}$ and $x_{b(n+1)} \in I_{[\alpha](bn+1)} \subseteq I_{[\alpha](n+1)}$ since $bn + 1 \geq n + 1$. Therefore α and $b_{\alpha,\beta}$ satisfy BW. □

To solve the functional interpretation of BW we must produce a witness for α and b in

$$\forall (x_i), \psi^{\mathbb{B}^{\mathbb{N}}\times\mathbb{N}^{\mathbb{N}}\to\mathbb{N}}\exists \alpha, b \forall n \leq \psi(\alpha, b)(bn < b(n+1) \wedge x_{bn} \in I_{[\alpha](n)}), \tag{7.6}$$

a statement which can be rephrased as

Finitary Bolzano-Weierstrass principle for $[0,1]_{\mathbb{Q}}$ **(Fin-BW).** Suppose (x_i) is a sequence of rationals in $[0,1]$. For an arbitrary functional $\psi\colon \mathbb{B}^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$, there exists $\alpha \in [0,1]$ and a function $b\colon \mathbb{N} \to \mathbb{N}$ such that $x_{b0}, x_{b1}, \ldots, x_{b(\psi(\alpha,b))}$ is a finite subsequence of (x_i) satisfying $|x_{bn} - \alpha| \leq 2^{-n}$ for all $n \leq \psi(\alpha, b)$.

A realizer for this computational interpretation of BW can be easily constructed from our realizer of Fin-Σ^0_1-WKL.

Theorem 7.7. *Given* (x_i) *and* ψ*, primitive recursively define* $g\colon \mathbb{N} \to \mathbb{N}$ *and* $\omega_\psi\colon \mathbb{B}^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ *by*

$$g(n) := n+1$$
$$\omega_\psi(\alpha,\beta) := b_{\alpha,\beta}(\psi(\alpha, b_{\alpha,\beta}))$$

where $b_{\alpha,\beta}$ *is defined by*

$$\begin{aligned} b_{\alpha,\beta}(0) &:= 0 \\ b_{\alpha,\beta}(n+1) &:= \mu i \in [b(n)+1, \beta(b(n)+1))(x_i \in I_{[\alpha](b(n)+1)}), \text{ else } b(n)+1. \end{aligned}$$

Construct $\beta_\psi := \beta_{\omega_\psi, g}$ *and* $\alpha_\psi := \alpha_{\omega_\psi, g}$ *as in Theorem 7.5. Then* $b_\psi := b_{\alpha_\psi,\beta_\psi}$ *and* α_ψ *realize* Fin-BW *(7.6).*

Proof. We have $T_0(h_{gn}n, gn)$ for all n since x_n lies in at least one I_s for $s\colon \mathbb{B}^n$. Therefore by Theorem 7.5

$$\forall n \leq \omega_\psi(\alpha_\psi, \beta_\psi) \exists i \in [n, \beta_\psi(n))(x_i \in I_{[\alpha_\psi](n)}).$$

Clearly $x_{b_\psi 0} = x_0 \in I_{\langle\rangle}$, and $n < \psi(\alpha_\psi, b_\psi)$ implies that $b_\psi(n) + 1 \leq b_\psi(\psi(\alpha_\psi, b_\psi)) = \omega_\psi(\alpha_\psi, \beta_\psi)$ (since we have $b_\psi(i) < b_\psi(i+1)$ by definition) and hence

$$\exists i \in [b_\psi(n)+1, \beta(b_\psi(n)+1))(x_i \in I_{[\alpha_\psi](b_\psi(n)+1)}).$$

This implies that $x_{b_\psi(n+1)} \in I_{[\alpha_\psi](b_\psi(n)+1)} \subseteq I_{[\alpha_\psi](n+1)}$ and we're done. □

Because our realizer for Fin-BW is constructed directly from that of Fin-Σ^0_1-WKL, it also inherits the game-semantics we have imposed on the latter, and its main component is the computation of an optimal strategy in the game $\mathcal{G}[\omega_\psi, g]$ for ω_ψ and g as defined in Theorem 7.7. We will not reiterate in detail the description of our realizer given in Section 7.1, although it is instructive to describe in a more informal manner the concrete sequential game that arises in this case.

Intuitively, in this scenario objects $\alpha\colon \mathbb{B}^{\mathbb{N}}$ represent real numbers in $[0,1]$ and objects $\beta\colon \mathbb{N}^{\mathbb{N}}$ search for convergent subsequences of (x_i): indeed, if α is an infinite branch of T witnessed by β then the function $b_{\alpha,\beta}$ yields a subsequence (x_{bi}) that converges to α. In order to construct an approximation $x_{b_\psi 0}, \ldots, x_{b_\psi(\psi(\alpha_\psi, b_\psi))}$ to a convergent subsequence we require an approximation α_ψ, β_ψ to an infinite branch of T of size $\omega_\psi \alpha_\psi \beta_\psi = b_\psi(\psi(\alpha_\psi, b_\psi))$ for $b_\psi = b_{\alpha_\psi, \beta_\psi}$: Because our g witnesses that T is infinite, the approximation satisfies

$$\forall n \leq b_\psi(\psi(\alpha_\psi, b_\psi)) \exists i \in [n, \beta_\psi(n))(x_i \in I_{[\alpha_\psi](n)}),$$

and in turn this guarantees that the construction b_ψ always satisfies $x_{b_\psi(n)} \in I_{[\alpha_\psi](n)}$ for $n \leq \psi(\alpha_\psi, b_\psi)$, and therefore provides us with a subsequence of length $\psi(\alpha_\psi, b_\psi)$ approximately converging to α_ψ.

The main instance of EPS in our realizer for Σ^0_1-WKL finds an optimal play in the game in which the selection function ε_n is responsible for collecting information about which intervals I_s for $|s| = n$ are inhabited by elements of (x_i) for $i \geq n$. Very roughly, the backtracking implemented by EPS is based on the following idea: an initial segment of a play β corresponds to a collection of candidates for the initial segment of a convergent subsequence, and the bar recursion attempts to extend this collection to an approximation to a convergent subsequence. If this extension lies outside of the intervals inhabited by the initial collection, we update our initial collection with elements from this extension and repeat the process.

Of course, further work is needed to make all this precise. However, at the very least we have been able to characterise the basic mathematical meaning of the main components of $\mathcal{G}[\omega_\psi, g]$, which we summarise in Table 8.2.

$\mathcal{G}[\omega_\psi, g]$	Finitary Bolzano-Weierstrass theorem
$g(n)$	there is an interval in partition of size 2^n that is inhabited by x_n
ε_n	find intervals in partition of size 2^n that are inhabited by some x_i for $i \geq n$
EPS^ε	construct approximation to convergent subsequence (via $b_{\alpha,\beta}$)
ω_ψ	make sure length of resulting approximation is large enough relative to ψ

Table 7.2: Functional interpretation of BW

7.3 Ramsey's theorem for pairs

We conclude this chapter with a computational interpretation of the proof of Ramsey's theorem for pairs by Erdős and Rado [25, Section 10.2]. Unsurprisingly, Ramsey's theorem has been extensively studied in logic, and several constructive interpretations have been offered. A computational analysis of Ramsey's theorem is given in e.g. [7, 21, 93], and also in [56] where a realizer for its ND interpretation based on the Erdős-Rado proof is given, even for $\mathsf{RT}^2_{<\infty}$. However, our work here is directly inspired by that of Kohlenbach and Kreuzer [57] who demonstrate that the Erdős-Rado proof can be formalised with Σ^0_1-WKL. Using a slightly different formalisation to theirs, we apply our realizer for Σ^0_1-WKL to extract a new computational interpretation of Ramsey's theorem in terms of the product of selection functions. Note that the primary motivation of [57] is to establish conservation results and bounds on the computational complexity of programs extracted from proofs using Ramsey's theorem (see also [55, 58]). Here we simply wish to gain some insight into the constructive meaning of the well-known and proof theoretically non-trivial Erdős-Rado construction.

A more thorough treatment of the interpretation of Ramsey's theorem using the product of selection functions is given by the author in [68]. Here, Howard's realizer for WKL is replaced by a binary product of selection functions (see Section 7.4), and using in addition the finite product of selection functions to interpret the infinite pigeonhole principle a computational interpretation of Ramsey's theorem is given purely in terms of three coordinated games.

Ramsey's theorem for pairs (RT^2_n). For any n-colouring $c\colon \mathbb{N}^{(2)} \to [n]$ of pairs of natural numbers, there exists an infinite set $X \subseteq \mathbb{N}$ that is pairwise monochromatic.

Pairwise monochromatic means that there exists some $x < n$ such that $c\{i,j\} = x$ for all $\{i,j\} \in X^{(2)}$. In our analysis of Ramsey's theorem we restrict ourselves to the two-colour case RT^2_2, although everything that follows can be directly extended to the general case (we will indicate how as we go along) and the behaviour of the general program is largely analogous to this special case. Written formally

$$\mathsf{RT}^2_2 \ : \ \forall c^{\mathbb{N}^{(2)} \to [2]} \exists x < 2, F^{\mathbb{N} \to \mathbb{N}} \forall n (Fn \geq n \wedge \forall i,j \leq n (Fi < Fj \to c(\{Fi, Fj\}) = x)). \tag{7.7}$$

Note that the infinite monochromatic set is encoded by F as $X := \{n \mid \exists k (Fk = n)\}$.

Notation. In the following it is more natural to encode c as a function $c\colon \mathbb{N}^2 \to [2]$ satisfying $c(i,j) = c(j,i)$ for $i < j$ and $c(i,i) = 0$.

The main idea behind the Erdős-Rado proof of RT^2_n is to organise the natural numbers into a tree (described as an ordering $\prec$ on $\mathbb{N}$) whose branches are *min-monochromatic*, in the sense that $c(i,j) = c(i,k)$ for $i \prec j \prec k$. This is the so-called Erdős-Rado (E/R) tree. The E/R tree is finitely branching, and therefore contains an infinite min-monochromatic branch $a\colon \mathbb{N}^{\mathbb{N}}$. By the infinite pigeonhole principle applied to the colouring $c^a(i) = c(a(i), a(i+1))$ there exists an infinite subset of this branch that is pairwise monochromatic. By encoding branches of the E/R tree as branches of a binary Σ^0_1-tree, we show that Ramsey's theorem can be derived from Σ^0_1-WKL.

It is important to remark why - as in [57] - the Erdös-Rado proof is preferred to Ramsey's seemingly simpler proof in [75]. Ramsey constructs an infinite min-monochromatic branch directly using dependent choice: First we define $a0 = 0$, then we use IPHP to produce an infinite set $A_1 \subseteq \mathbb{N}\backslash 0$ that is monochromatic under $c_0(i) = c(0,i)$ and define $a(1) = \min A_1$. Next use IPHP to produce an infinite set $A_2 \subseteq A_1\backslash a(1)$ that is monochromatic under $c_{a(1)}(i) = c(a(1),i)$ and define $a(2) = \min A_2$ and so on. It is easy to see that the resulting a is min-monochromatic. However, Ramsey's construction uses dependent choice of type 1 (and cannot even be formalised in the subsystem ACA_0 [84]), therefore its computational interpretation would seemingly involve bar recursion of level 1. Our interpretation of the Erdős-Rado proof, on the other hand, makes use of the product of selection functions of lowest type only, meaning that our construction is computationally simpler.

Definition 7.8 (Erdős/Rado tree). Given a colouring $c \colon \mathbb{N}^2 \to [2]$, define a partial order $\prec$ on $\mathbb{N}$ recursively as follows:

1. $0 \prec 1$

2. Given that $\prec$ is already defined on the initial segment of the natural numbers $[m]$, for $i < m$ define

$$i \prec m \quad \text{iff} \quad c(k,i) = c(k,m), \text{ for all } k \prec i.$$

Note that $\prec$ is a suborder of $\leq$, and also that $\prec$ is decidable.

It is not too hard to show that $\prec$ is transitive, and also that $i \prec j$ iff $i < j$ for $i,j \in \mathrm{pd}(m)$. Therefore $\prec$ defines a tree on $\mathbb{N}$, the so-called Erdős/Rado tree, whose branches are min-monochromatic i.e. $c(k,i) = c(k,j)$ for $k \prec i \prec j$. Moreover, this tree is binary branching because i and j are distinct successors of k if and only if $c(k,i) \neq c(k,j)$ (for the general n-colour case the E/R-tree is n-branching). For proofs of these facts see [57, Section 4].

Lemma 7.9. *The tree $\prec$ has an infinite branch, or in other words there exists $a \colon \mathbb{N} \to \mathbb{N}$ satisfying*

$$\forall n (n \leq an \wedge \forall i,j,k < n(ak < ai \wedge ak < aj \to c(ak,ai) = c(ak,aj))), \tag{7.8}$$

provably in $\mathsf{WE\text{-}PA}^\omega + \mathsf{QF\text{-}AC} + \Sigma^0_1\text{-}\mathsf{WKL}$.

Proof. Define the Σ^0_1 tree T by

$$T(s) :\equiv \exists k (\underbrace{|s| \leq k \wedge \forall i < |s| (s_i = 0 \leftrightarrow i \prec k)}_{T_0(s,k)}),$$

Note that $T(s * t, k) \to T(s,k)$. If a binary sequence s belongs to T then it is the characteristic function of a proper initial segment of a branch in the E/R tree. Clearly T

has branches of arbitrary length: for any n we have $T_0(s,n)$ where s is the branch of length n defined by $s_i = 0 \leftrightarrow i \prec n$, i.e. s is the characteristic function of $\mathrm{pd}(n)$. Therefore by Σ^0_1-WKL there exist α, β satisfying

$$\forall n \exists k \in [n, \beta(n)) \forall i < n(\alpha(i) = 0 \leftrightarrow i \prec k). \tag{7.9}$$

We must now show that α encodes an infinite min-monochromatic branch. Let us construct the function $a_{\alpha,\beta} \colon \mathbb{N} \to \mathbb{N}$ as

$$a(n) = \begin{cases} 0 & \text{if } n = 0 \\ \mu i \in [n, \beta^{(2)}(n)) \text{ such that } \alpha(i) = 0, \text{ else } n & \text{if } n > 0. \end{cases} \tag{7.10}$$

We claim that a satisfies $\alpha(a(n)) = 0$ for all n. If this is so, then a also satisfies (7.8), since $a(n) \geq n$ by definition, and if $ak < ai < aj$ then setting $n = aj + 1$ in (7.9) gives some k' such that $ai, aj, ak \prec k'$ and therefore $ak \prec ai \prec aj$ and hence $c(ak, ai) = c(ak, aj)$.

Clearly $\alpha(a(0)) = \alpha(0) = 0$ since $0 \prec k$ for all k. To prove the claim for $n > 0$, suppose for contradiction that $\alpha(i) \neq 0$ for all $i \in [n, \beta^{(2)}(n))$. Let $j < n$ be the largest such that $\alpha(j) = 0$ (this exists since $0 \prec k$ for all k). By (7.9) let $k_0 \in [n, \beta(n))$ be the least satisfying $\forall i < n(\alpha(i) = 0 \leftrightarrow i \prec k_0)$. Then k_0 is an immediate successor of j, otherwise if $j \prec x \prec k_0$ then $x \geq n$ by maximality of j, contradicting minimality of k_0. Next, define $k_1 \in [\beta(n), \beta^{(2)}(n))$ to be the least satisfying $\forall i < \beta(n)(\alpha(i) = 0 \leftrightarrow i \prec k_1)$. By assumption, j is the greatest number $< \beta(n)$ with $\alpha(j) = 0$, therefore k_1 is also an immediate successor of j. But there exists $k_2 \geq \beta^{(2)}(n)$ satisfying $\forall i < \beta^{(2)}(n)(\alpha(i) = 0 \leftrightarrow i \prec k_2)$. Because $k_0 < k_1 < k_2$ and $j \prec k_2$, by the fact that the E/R tree is binary branching we must have $k_i \prec k_2$ and therefore $\alpha(k_i) = 0$ for $i = 0$ or 1. But $k_i \in [n, \beta^{(2)}(n))$, a contradiction. □

Remark 7.10. In generalising the proof of Lemma 7.9 for m-colours we would need to define $a(n) = \mu k \in [n, \beta^{(m)}) \ldots$ to reflect the fact that the E/R-tree is m-branching.

Remark 7.11. The formalisation of Ramsey's theorem in [57] is based on a slightly different Σ^0_1-tree whose branches represent colourings of branches of the E/R tree as opposed to characteristic functions of those branches. There the priority is a formalisation that is optimal with respect to the amount of induction used, here we have no such concerns and prefer a formalisation in which branches of T directly encode branches of the E/R tree.

Theorem 7.12. RT^2_2 *is instancewise provable from* Σ^0_1-WKL*, over* $\mathsf{WE\text{-}PA}^\omega + \mathsf{QF\text{-}AC}$.

Proof. We use Lemma 7.9 along with IPHP, which we recall states that for any colouring $c \colon \mathbb{N} \to [k]$ there exists $x \leq k$ such that

$$\forall n \exists m(m \geq n \wedge c(m) = x).$$

Define the colouring $c^a \colon \mathbb{N} \to [2]$ as $c^a(n) := c(a(n), a(a(n)+1))$ for a as defined in Lemma 7.9. Then by $\mathsf{QF\text{-}AC}$ there exists $x < 2$ and $p \colon \mathbb{N} \to \mathbb{N}$ such that $p(n) \geq n$ for all n and

$$x = c^a(pi) = c(a(pi), a(a(pi)+1)) \overset{(7.8)}{=} c(a(pi), a(pj))$$

for any i, j with $a(pi) < a(pj)$. Hence setting $F(n) := a(pn)$ we have

$$F(n) \geq n \wedge \forall i, j \leq n(Fi < Fj \to c(Fi, Fj) = x),$$

for arbitrary n, and therefore F and x satisfy (7.7). □

7.3.1 A bar recursive interpretation of RT^2_2

The functional interpretation of RT^2_2 is given by

$$\forall c, \eta^{[2] \times \mathbb{N}^{\mathbb{N}} \to \mathbb{N}} \exists x, F \forall n \leq \eta_x F(Fn \geq n \wedge \forall i, j \leq n(Fi < Fj \to c(Fi, Fj) = x)). \tag{7.11}$$

which can be informally stated as

Finitary Ramsey's theorem for pairs ($\mathsf{Fin\text{-}RT}^2_2$). Given a colouring $c \colon \mathbb{N}^{(2)} \to [2]$ and a functional $\eta \colon [2] \times \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ there exists $x < 2$ and a function F such that the finite set $X := \{Fk \mid k \leq \eta_x F\}$ is pairwise monochromatic.

Here X forms a finite approximation to an infinite monochromatic set. Note that by defining $\tilde{F}$ by $\tilde{F}(0) = F(0)$ and $\tilde{F}(n+1) = F(\tilde{F}(n)+1)$ and setting $\tilde{\eta}_x F := \max_{i \leq \eta_x \tilde{F}}(\tilde{F}(i)+1)$ we obtain

$$\forall c, \eta^{[2] \times \mathbb{N}^{\mathbb{N}} \to \mathbb{N}} \exists x, \tilde{F} \forall m < n \leq \eta_x \tilde{F}(\tilde{F}(m) < \tilde{F}(n) \wedge c(\tilde{F}(m), \tilde{F}(n)) = x)$$

from (7.11), and so we can always produce an approximation $\tilde{F}$ of *size* $\eta_x \tilde{F}$.

The following lemma is the computational interpretation of Lemma 7.9.

Lemma 7.13. *Given an arbitrary functional $\psi \colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ define $g \colon \mathbb{N} \to \mathbb{N}$ and $\omega \colon \mathbb{B}^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ by*

$$\begin{aligned} g(n) &:= n+1 \\ \omega_\psi(\alpha, \beta) &:= \max_{i \leq \psi(a_{\alpha,\beta})} (\max\{i, \beta(i), \beta^{(2)}(i)\}) \end{aligned}$$

where $a_{\alpha,\beta}$ is as in Lemma 7.9. Construct $\beta_\psi = \beta_{\omega_\psi, g}$ and $\alpha_\psi = \alpha_{\omega_\psi, g}$ as in Theorem 7.5. Then $a_\psi = a_{\alpha_\psi, \beta_\psi}$ satisfies

$$\forall n \leq \psi a_\psi(n \leq a_\psi n \wedge \forall i, j, k \leq n(a_\psi k < a_\psi i \wedge a_\psi k < a_\psi j \to c(a_\psi k, a_\psi i) = c(a_\psi k, a_\psi j))). \tag{7.12}$$

Proof. We have $T_0(s,n)$ for the branch $s \in \mathbb{B}^{(n)}$ given by $s_i = 0 \leftrightarrow i \prec n$, therefore we have $T_0'(h_n n, gn)$ for all n. Then by Theorem 7.5 we have

$$\forall n \leq \omega_\psi(\alpha_\psi, \beta_\psi) \exists k \in [n, \beta_\psi(n)) \forall i < n(\alpha_\psi(i) = 0 \leftrightarrow i \prec k). \tag{7.13}$$

Now, $a_\psi n \geq n$ by definition. Next, we can show that $\alpha(a_\psi(n)) = 0$ for all $n \leq \psi(a_\psi)$ by definition of ω_ψ and (7.13), using the same argument as in the proof of Lemma 7.9.

Finally, for $i, j, k \leq \psi a_\psi$ with (w.l.o.g) $a_\psi k < a_\psi i < a_\psi j$, by (7.13) for $n = a_\psi j + 1 \leq \beta_\psi^{(2)}(j) \leq \omega_\psi(\alpha_\psi, \beta_\psi)$ and the fact that $\alpha_\psi(a_\psi i) = \alpha_\psi(a_\psi j) = \alpha_\psi(a_\psi k) = 0$ there is some k' such that $a_\psi k, a_\psi i, a_\psi j \prec k'$ and therefore $a_\psi k \prec a_\psi i \prec a_\psi j$, so we must have $c(a_\psi k, a_\psi i) = c(a_\psi k, a_\psi j)$ by definition of $\prec$. Thus a_ψ satisfies (7.12). □

Now we interpret the instance of IPHP in the proof of RT^2_2. One can very naturally interpret IPHP using a finite form of bar recursion such as the finite product of selection functions (see [66], [31, Section 7] for a general solution). In this instance we only require the binary product $\otimes$ (see Definition 5.2), while the n-colour case will involve the finite product of length n. We give the construction directly without further explanation, as it forms only a minor part of the overall realizer.

Lemma 7.14. *Suppose we have a colouring $d \colon \mathbb{N} \to [2]$ and a functional $\delta \colon [2] \times \mathbb{N}^\mathbb{N} \to \mathbb{N}$. Then there exists $x < 2$ and $p \colon \mathbb{N} \to \mathbb{N}$ dependent on d and δ satisfying*

$$\forall i \leq \delta_x p(pi \geq i \wedge d(pi) = x). \tag{7.14}$$

Proof. Define $\tilde{\delta}_x p := \mu i \leq \delta_x p \neg(pi \geq i \wedge d(pi) = x)$ and let $\langle a_0, a_1 \rangle = (\tilde{\delta}_0 \otimes \tilde{\delta}_1)(\max)$ and $N = \max\{a_0, a_1\}$. Then we have

$$a_0 = \tilde{\delta}_0 p_0 \qquad a_1 = \tilde{\delta}_1 p_1 \qquad N = p_0(a_0) = p_1(a_1)$$

where $p_1 := \lambda y.\max\{a_0, y\}$ and $p_0 := \lambda x.\max\{x, \tilde{\delta}_1(\lambda y.\max\{x, y\})\}$ (these are just Spector's equations (5.3) for the binary product of selection functions). Let $x_{d,\delta} = d(N)$ and $p_{d,\delta} = p_{x_{d,\delta}}$. Then x and p_x satisfy (7.14) because (suppressing parameters) $p_x(\tilde{\delta}_x p_x) = p_x(a_x) = N \geq a_x = \tilde{\delta}_x p_x$ and $d(p_x(\tilde{\delta}_x p_x)) = d(p_x(a_x)) = d(N) = x$. Therefore

$$p_x(\tilde{\delta}_x p_x) \geq \tilde{\delta}_x p_x \wedge d(p_x(\tilde{\delta}_x p_x)) = x$$

and so (7.14) must hold by definition of $\tilde{\delta}_i$. □

We are now ready to construct our realizer for $\mathsf{Fin\text{-}RT}^2_2$.

Theorem 7.15. *Suppose we are given $\eta \colon [2] \times \mathbb{N}^\mathbb{N} \to \mathbb{N}$ and $c \colon \mathbb{N}^2 \to [2]$. Suppressing these parameters, for arbitrary $a \colon \mathbb{N} \to \mathbb{N}$ define $\delta^a \colon [2] \times \mathbb{N}^\mathbb{N} \to \mathbb{N}$ by $\delta^a_x p{:=}\eta_x(a \circ p)$, define* *the colouring $c^a(i) := c(a(i), a(a(i)+1))$ and let $x_a\colon [2]$, $p_a\colon \mathbb{N} \to \mathbb{N}$ be constructed as in Lemma 7.14 on parameters c^a, δ^a so that they satisfy*

$$\forall i \leq \eta_{x_a}(a \circ p_a)(p_a(i) \geq i \wedge c^a(p_a(i)) = x_a). \tag{7.15}$$

Now define $\psi a := \max_{i \leq \eta_{x_a}(a \circ p_a)}(\max\{p_a(i), a(p_a(i))+1\})$ and define a_ψ as in Lemma 7.13 so that it satisfies

$$n \leq a_\psi n \wedge \forall i, j, k \leq n(a_\psi k < a_\psi i \wedge a_\psi k < a_\psi j \to c(a_\psi k, a_\psi i) = c(a_\psi k, a_\psi j)) \tag{7.16}$$

for all $n \leq \psi a_\psi$. Then $F_{\eta,c} = a_\psi \circ p_{a_\psi}$ and $x_{\eta,c} = x_{a_\psi}$ satisfy the functional interpretation of $\mathsf{RT}^2_2(c)$:

$$\forall n \leq \eta_{x_{\eta,c}} F_{\eta,c}(F_{\eta,c} n \geq n \wedge \forall i, j \leq n(F_{\eta,c} i < F_{\eta,c} j \to c(F_{\eta,c} i, F_{\eta,c} j) = x_{\eta,c})).$$

Proof. For $n \leq \eta_x F = \eta_{x_{a_\psi}}(a_\psi \circ p_{a_\psi})$ we have

$$Fn = a_\psi(p_{a_\psi} n) \overset{(7.16)}{\geq} p_{a_\psi} n \overset{(7.15)}{\geq} n,$$

and for $i, j \leq \eta_{x_{a_\psi}}(a_\psi \circ p_{a_\psi})$ and $Fi < Fj$ we have

$$x_{a_\psi} \overset{(7.15)}{=} c^{a_\psi}(p_{a_\psi}(i)) = c(a_\psi(p_{a_\psi}(i)), a_\psi(a_\psi(p_{a_\psi}(i))+1)) \overset{(7.16)}{=} c(\underbrace{a_\psi(p_{a_\psi}(i))}_{Fi}, \underbrace{a_\psi(p_{a_\psi}(j))}_{Fj}))$$

where note that for the last step we have $p_a(i), p_a(j), a(p_a(i))+1 \leq \psi a$ and $a(p_a(i)) < a(a(p_a(i))+1) \wedge a(p_a(i)) < a(p_a(j))$. This completes the proof. □

A map of the whole construction is given in Figure 7.2.

$$\dfrac{\overset{\text{L. 7.14}}{\lambda a\ .\ |\mathsf{IPHP}[c^a]^{\mathrm{N}}|^{p_a, x_a}_{\delta^a}} \qquad \dfrac{\overset{\text{Th. 7.5}}{|\Sigma^0_1\text{-}\mathsf{WKL}|^{\alpha_\psi, \beta_\psi}_{\omega_\psi}}}{|\mathsf{E/R}(c)|^{a_\psi}_{\psi_\eta}}\ \text{Lem. 7.13}}{|\mathsf{RT}^2_2(c)|^{a_\psi \circ p_{a_\psi}, x_{a_\psi}}_{\eta}}\ \text{Th. 7.15}$$

Figure 7.2: Constructive proof of $\mathsf{Fin}\text{-}\mathsf{RT}^2_2(c)$.

Once more, we will briefly describe the specific sequential game that arises from our computational interpretation of $\mathsf{RT}^2_2(2)$.

Branches s of our Σ^0_1-tree T represent proper initial segments of branches of the Erdős-Rado tree via their characteristic functions, and as proved in Lemma 7.9, *infinite* branches α of T encode infinite branches of the E/R tree, the witnessing function β allowing us to recursively find zeroes of the sequence α via the function $a_{\alpha,\beta}$.

In order to construct an approximation a_ψ to an infinite branch of the E/R tree of size ψ we require an approximation α_ψ, β_ψ to an infinite branch of our original tree T of size $\omega_\psi \alpha_\psi \beta_\psi = \max_{i \leq \psi a_\psi}(\max\{i, \beta_\psi(i), \beta_\psi^{(2)}(i)\})$. This approximation satisfies

$$\forall n \leq \omega_\psi(\alpha_\psi, \beta_\psi) \exists k \in [n, \beta_\psi(n)) \forall i < n (\alpha_\psi(i) = 0 \leftrightarrow i \prec k)$$

and this is sufficient to ensure that $a_\psi = a_{\alpha_\psi, \beta_\psi}$ always picks out an element of a min-monochromatic branch for $n \leq \psi(a_\psi)$, as proved in Lemma 7.13.

Now in turn, to obtain an approximation F to an infinite pairwise monochromatic set of size η, we require an approximation a_{ψ_η} to an E/R branch of size ψ_η (as defined in Theorem 7.15), where ψ_η is constructed using finite bar recursion to interpret IPHP (Lemma 7.14). This ensures that our approximation a_{ψ_η} to an infinite branch of the E/R tree is large enough to also contain an approximation to an min-monochromatic set relative to η. While the definitions of ω_ψ and ψ_η may seem quite complex, their purpose is simply to ensure that our approximation to α, β propagates through the constructions $a_{\alpha,\beta}$ and p_a to an approximation of F of sufficient quality.

The main instance of EPS in our realizer forms a sequential game in which the role of the selection function ε_n is to collect information about which branches of length n represent proper initial segments branches of the E/R tree, so in other words discover information about how the initial segment of the natural numbers $[n]$ is arranged in the E/R tree. By working together they are able to compute an approximation to an infinite branch of the E/R tree, and the backtracking implemented by EPS roughly corresponds to a backtracking along the E/R tree itself.

$\mathcal{G}[\omega_{\psi_\eta}, g]$	Finitary Ramsey's theorem
$g(n)$	there is a proper i.s. of a branch of the E/R tree in $[n]$ given by pd(n)
ε_n	find proper i.s.'s t of the E/R tree in $[n]$ s.t. $t \prec \text{pd}(k)$ for $k \geq n$
EPS^ε	construct an approximation to infinite branch of E/R tree
ω_{ψ_η}	ensure approximation contains infinite monochromatic set relative to η

Table 7.3: Functional interpretation of RT^2_2

7.4 An alternative to Howard's realizer of WKL

We have focused thus far on making clear the game-theoretic intuition behind the main instance of bar recursion in the realizer of Σ^0_1-WKL. However, it is also possible to replace Howard's realizer for WKL via a binary form of GBR with a binary version of EPS instead, giving us a game-theoretic computational interpretation of WKL as well.

Notation. Given a decidable binary tree T, let us define the tree $T_s(t) := T(s * t)$. We write $\mathsf{Depth}_n(T) :\equiv \exists s(|s| = n \wedge T(s))$.

Lemma 7.16. *The selection function* $\varepsilon\colon \mathbb{B}^* \to J_{\mathbb{N}}\mathbb{B}$ *given by*

$$\varepsilon_s(p) \overset{\mathbb{B}}{:=} \begin{cases} 0 & \textit{if } \mathsf{Depth}_{p(0)+1}(T_s) \to \mathsf{Depth}_{p(0)}(T_{s*0}) \\ 1 & \textit{otherwise} \end{cases} \tag{7.17}$$

satisfies

$$\forall s, p(\mathsf{Depth}_{p(\varepsilon_s p)+1}(T_s) \to \mathsf{Depth}_{p(\varepsilon_s p)}(T_{s*\varepsilon_s p})).$$

Proof. If $\mathsf{Depth}_{p(0)+1}(T_s) \to \mathsf{Depth}_{p(0)}(T_{s*0})$ then $\varepsilon_s p = 0$ and (7.17) is clearly true. So let us assume that $\mathsf{Depth}_{p(0)+1}(T_s) \wedge \neg\mathsf{Depth}_{p(0)}(T_{s*0})$ so that $\varepsilon_s p = 1$, and suppose that the premise of (7.17) i.e. $\mathsf{Depth}_{p(1)+1}(T_s)$ is true. We want to show that $\mathsf{Depth}_{p(1)}(T_{s*1})$. There are two cases to consider.

Case (a): $p(0) \geq p(1)$, and we have

$$\mathsf{Depth}_{p(0)+1}(T_s) \wedge \neg\mathsf{Depth}_{p(0)}(T_{s*0}) \to \mathsf{Depth}_{p(0)}(T_{s*1}) \to \mathsf{Depth}_{p(1)}(T_{s*1}).$$

Case (b): $p(0) < p(1)$, and we have $\neg\mathsf{Depth}_{p(0)}(T_{s*0}) \to \neg\mathsf{Depth}_{p(1)}(T_{s*0})$. But $\mathsf{Depth}_{p(1)+1}(T_s)$, and therefore $\mathsf{Depth}_{p(1)}(T_{s*1})$. □

Theorem 7.17. *Given* $\varphi\colon \mathbb{B}^{\mathbb{N}} \to \mathbb{N}$, *define* $\alpha_\varphi := \mathsf{EPS}^{\varepsilon, q^\varphi, \varphi}_{\mathbb{B},\mathbb{N}}$ *where* ε *is defined as in Lemma 7.16, and* $q^\varphi(\alpha) := \varphi\alpha - k - 1$ *for the least* $k < \varphi\alpha$ *refuting*

$$\forall k < \varphi\alpha(\mathsf{Depth}_{\varphi\alpha-k}(T_{[\alpha](k)}) \to \mathsf{Depth}_{\varphi\alpha-k-1}(T_{[\alpha](k+1)})). \tag{7.18}$$

Then α_φ *and* $n_\varphi := \varphi\alpha_\varphi$ *satisfy*

$$\exists s(|s| = n \wedge T(s)) \to T([\alpha](\varphi\alpha))$$

and therefore realize the functional interpretation of WKL.

Proof. Suppressing the parameter on α_φ, by Spector's equations, substituting $s = [\alpha](\varphi\alpha - q^\varphi\alpha - 1)$, $p = p_s$ into (7.17) we have

$$\mathsf{Depth}_{q^\varphi\alpha+1}(T_{[\alpha](\varphi\alpha-q^\varphi\alpha-1)}) \to \mathsf{Depth}_{q^\varphi\alpha}(T_{[\alpha](\varphi\alpha-q^\varphi\alpha)}).$$

But this implies (7.18) must hold by definition of q^φ. Now $\exists s(|s| = n_\varphi \wedge T(s)) \to \mathsf{Depth}_{\varphi\alpha}(T_{\langle\rangle})$, and by induction on (7.18) from $k = 0$ to $k = \varphi - 1$ we have

$$\mathsf{Depth}_{\varphi\alpha}(T_{\langle\rangle}) \to \mathsf{Depth}_0(T_{[\alpha](\varphi\alpha)}).$$

But $\mathsf{Depth}_0(T_{[\alpha](\varphi\alpha)}) \to T([\alpha](\varphi\alpha))$ and we're done. □

This computational interpretation of WKL corresponds to a binary game. The strategy ε_s at position s defined by the selection functions given in Lemma 7.16 is to pick a boolean b such that if s extends to a branch in T of length $|s| + p(b) + 1$ then $s * b$ also extends to a branch of length $|s| + p(b) + 1$.

Given φ, by choosing q^φ suitably as in Theorem 7.17, the optimal play of the game determined by the ε_n is a sequence α_φ such that for all $k \leq \varphi\alpha_\varphi$, whenever $[\alpha_\varphi](k)$ extends to a branch of length $\varphi\alpha_\varphi$, so does $[\alpha_\varphi](k+1)$. If T is infinite then $\langle\rangle$ extends to a branch of length $\varphi\alpha_\varphi$. Hence, by induction the relevant part $[\alpha_\varphi](\varphi\alpha_\varphi)$ of this optimal play must be in T^β, and is therefore an approximation to an infinite branch.

This interpretation of WKL is certainly more complex than Howard's realizer, and in particular to verify that it works one makes use of the property that T is prefix closed in several places (which means that using it in our realizer for Σ^0_1-WKL would require a more carefully calibrated approximation to the characteristic function β). However, it has the advantage that when used instead of the usual binary bar recursion one obtains computational interpretations of Σ^0_1-WKL and therefore also Bolzano-Weierstrass theorem and Ramsey's theorem that can be described purely in terms of sequential games. For the latter in particular, given that the finite product of selection functions is used to interpret IPHP, we obtain the correspondence

$$\begin{aligned} \Pi^0_1\text{-countable choice} \quad &\mapsto \quad \mathsf{EPS}_{\mathbb{N},\mathbb{N}} \text{ (unbounded sequential game)} \\ \text{weak König's lemma} \quad &\mapsto \quad \mathsf{EPS}_{\mathbb{B},\mathbb{N}} \text{ (binary unbounded sequential game)} \\ \text{infinite pigeonhole principle} \quad &\mapsto \quad \mathsf{EPS}^{\text{fin}}_{\mathbb{N},\mathbb{N}} \text{ (finite sequential game)} \end{aligned}$$

and so we are able to produce a realizer for the functional interpretation of RT^2_n purely in terms of three symbiotic games, each representing one of the non-trivial axioms in its formal proof. This construction, along with a description of the realizer, is given in full in [68].

CHAPTER

EIGHT

The Minimal-Bad-Sequence Argument

In our final case study we investigate the computational content of the 'minimal-bad-sequence' argument, a consequence of dependent choice best known for its use in Nash-Williams' proof of Higman's lemma and Kruskal's theorem [64]. Our main contribution is the construction of a bar recursive realizer for Gödel's functional interpretation of the minimal-bad-sequence argument over arbitrary well-founded relations. We analyse the behaviour of our program, and then suggest a different approach based on Berger's realizability interpretation of the classically equivalent principle of open induction. We conclude by applying our realizer to extract a new bound for Higman's lemma.

The minimal-bad-sequence argument (MBS) is of central importance in the theory of well-quasi-orders, and as such has become a focal point of research into computational aspects of classical reasoning used in combinatorics. The constructive content of the minimal-bad-sequence argument has been widely analysed (see for instance [20, 91]), and programs have been extracted from its classical proof using a variety of formal methods including the A-translation [63] and inductive definitions [23]. An extensive study of MBS has been carried out by Berger and Seisenberger [13, 82], who extend and improve the aforementioned techniques and implement them in the Minlog proof assistant. We make no effort to provide an exhaustive list of the research that has been done in this area (for a more detailed account see [82]), but we must be careful to highlight the motivating factors behind the work in this chapter.

To begin with, our constructive proof of the minimal-bad-sequence argument is new, and the first based on Gödel's functional interpretation. Moreover, to our knowledge it constitutes one of the first applications of the functional interpretation in the theory of well-quasi-orderings, and as such our interpretation may pave the way for new applications of proof theory in combinatorics - although admittedly this is not our primary goal. Nevertheless, we give a practical illustration of how our realizer can be used to witness Higman's lemma and bound the length of bad sequences of words over arbitrary decidable well-quasi-orders.

More importantly, as in previous chapters our aim is to produce a realizer that can be feasibly understood on a human level and which provides us with genuine insight into the constructive meaning of mathematical proofs. The substantial logical complexity of the minimal-bad-sequence argument (which in the language of second order arithmetic is equivalent to Π^1_1-CA_0 [62]) means that such practical program extraction poses quite a challenge. Indeed, many existing case studies are essentially a formalisation of the proof of the minimal-bad-sequence argument, the program extraction itself being carried out with the help of a proof assistant. This approach often results in output that is not easy to comprehend (one of the first realizers obtained by Murthy [63] ran to 12MB of computer code). In contrast, our interpretation is carried out by hand and yields a realizer in terms of the product of selection functions that is somewhat easier to describe in qualitative terms.

The minimal-bad-sequence argument is an interesting case study for the extraction of programs from proofs because it is classically equivalent to the principle of *open induction*: induction over the lexicographic ordering on sequences restricted to open predicates. As demonstrated by Berger [10], a concise realizability interpretation of open induction is naturally given by *open recursion* over the lexicographic ordering, and this provides us with a direct alternative means of extracting programs from classical proofs based on MBS with which we can avoid the route through dependent choice and bar recursion. We discuss the possibility of a similar technique in the context of the functional interpretation, and suggest an alternative to Spector's bar recursion based on open recursion that can be used when interpreting proofs based on the minimal-bad-sequence argument.

8.1 The minimal bad sequence construction

Suppose that we are given a decidable binary relation $<\colon \rho \times \rho \to \mathbb{B}$ that is provably well-founded - in other words the schema of transfinite induction $\mathsf{TI}_<$ over $<$

$$\mathsf{TI}_< \ : \ \forall y^\rho(\forall z < yA(z) \to A(y)) \to \forall x A(x)$$

for arbitrary A is valid in our formal system. Note that transfinite induction is classically equivalent to the least element principle over $<$ given by

$$\mathsf{LEP}_< \ : \ \exists x A(x) \to \exists y(A(y) \wedge \forall z < y \neg A(z))$$

The lexicographic ordering over $<$ is a binary relation on infinite sequences of type $\rho^{\mathbb{N}}$ defined by

$$u <_{\mathrm{lex}} v :\equiv \exists n([u](n) = [v](n) \wedge u(n) < v(n)).$$

While the relation $<_{\mathrm{lex}}$ is neither decidable nor well-founded (apart from trivial cases), transfinite induction over $<_{\mathrm{lex}}$ is valid provided we restrict ourselves to a certain class of formulas called the *open formulas.* In the presence of classical logic, an open formula is a predicate on sequences of type $\rho^{\mathbb{N}}$ that is equivalent to one of the form

$$B(u) \equiv \forall n C([u](n)) \to \exists n D([u](n))$$

for C and D arbitrary predicates on ρ^*. The schema of open induction is given by

$$\mathsf{OI}_{\rho,<} \ : \ \forall v^{\rho^{\mathbb{N}}}(\forall w{<_{\mathrm{lex}}}v B(w) \to B(v)) \to \forall u B(u)$$

where B ranges over open formulas. Open induction was formulated by Raoult in [76], and is investigated from an intuitionistic perspective in [10, 20], where one typically has to work with a restricted definition of open formulas: for instance in [20] an open formula is simply defined to be a predicate of the form $B(u) \equiv \exists n B_0([u](n))$ where B_0 is a decidable.

Example 8.1. The open predicate on sequences of words which we use in Section 8.4 to prove Higman's lemma is given by

$$B(u^{(X^*)^{\mathbb{N}}}) :\equiv \exists n(\underbrace{\exists i < j < n(u_i \leq_{X^*} u_j)}_{B_0([u](n))})$$

for some well-quasi-ordering $(X, \leq_X)$.

We call a sequence u *bad* relative to an open predicate B if $\neg B(u)$ holds (and good otherwise). Open induction is classically equivalent to the so-called minimal bad sequence argument

$$\mathsf{MBS}_{\rho,<} \ : \ \exists u \neg B(u) \to \exists v(\neg B(v) \wedge \forall w{<_{\mathrm{lex}}}v B(w))$$

which is the statement that whenever a bad sequence u exists, then a minimal bad sequence v exists relative to $<_{\mathrm{lex}}$. The existence of a minimal bad sequence follows classically from $\mathsf{LEP}_<$ and dependent choice, via a famous construction due to Nash-Williams.

Theorem 8.2 (Nash-Williams [64])**.** *Dependent choice proves the minimal-bad-sequence argument.*

Proof. Suppose that $B(u) = \forall n C([u](n)) \to \exists n D([u](n))$ is open, and that there exists a bad sequence u (i.e. satisfying $\neg B(u)$). Construct a sequence v recursively as follows:

1. By the least element principle on $<$, choose a minimal element $v_0 \colon \rho$ such that v_0 extends to a bad sequence, but all sequences extending y for $y < v_0$ are good.

2. Given $v_0, \ldots, v_{n-1}$, choose v_n such that $v_0, \ldots, v_{n-1}, v_n$ extends to a bad sequence, but all sequences extending $v_0, \ldots, v_{n-1}, y$ for $y < v_n$ are good.

By dependent choice we obtain an infinite sequence v such that w is good for any $w{<_{\mathrm{lex}}}v$. But for each n, $[v](n)$ extends to a bad sequence, so by classical logic we must have in particular $C([v](n)) \wedge \neg D([v](n))$, and therefore $\forall n(C([v](n)) \wedge \neg D([v](n)))$ which is equivalent to $\neg B(v)$, so v itself is a bad sequence. □

In the main part of this chapter we extract bar recursive realizers for the functional interpretation of MBS by analysing the computational content of Nash-Williams' proof. Our first step is to restate Nash-Williams' construction in a more logically explicit form, which we give as Theorem 8.4. Note that we do not consider this to be a full formalisation of the minimal-bad-sequence construction (in the sense of e.g. [82]) - this would be unnecessary here as we extract our realizer 'by hand'. Our formalisation acts only as a *guide* for how to proceed in the program extraction.

Remark 8.3. (i) Because we impose no restriction on the type ρ in $\mathsf{OI}_{\rho,<}$, for higher-types ρ the predicate $<_{\rm lex}$ contains additional quantifiers belonging to $=_{\rho^*}$ (this is slightly different to the situation in [10, 82] who work in a neutral variant of PA^ω where higher-type equality is taken as primitive). To address this, we note that over $\mathsf{E}\text{-}\mathsf{HA}^\omega$ we have

$$\forall w{<_{\rm lex}}vB(w) \leftrightarrow [\forall w, n(w(n) < v(n) \to B([v](n) @ w))]$$

so we will reformulate MBS as

$$\exists u \neg B(u) \to \exists v(\neg B(v) \wedge \forall w, n(w(n) < v(n) \to B([v](n) @ w))).$$

(ii) We extract a realizer for MBS for the *restricted* class of open predicates defined in [20] to be those of the form $B(u) \equiv \exists n B_0([u](n))$ for B_0 quantifier-free. This form of MBS is sufficient for many applications, including Higman's lemma (see Section 8.4).

(iii) In our formal proof we will use the sequential variant of dependent choice $\mathsf{DC}_{\mathsf{seq}}$ introduced in Section 5.3.

Theorem 8.4. $\mathsf{WE}\text{-}\mathsf{PA}^\omega + \mathsf{QF}\text{-}\mathsf{AC} + \mathsf{DC}_{\mathsf{seq}} \vdash \mathsf{MBS}_{\rho,<}$ *for any decidable relation such that* $\mathsf{TI}_<$ *is provable in* $\mathsf{WE}\text{-}\mathsf{PA}^\omega + \mathsf{QF}\text{-}\mathsf{AC} + \mathsf{DC}_{\mathsf{seq}}$.

Proof. There are three main parts to our formalisation of the minimal-bad-sequence construction: 1. Establishing the inductive step via the least element principle; 2. Using dependent choice to construct a minimal bad sequence; 3. Verifying that this sequence has the required properties.

For the inductive step, suppose we are given a finite sequence $s\colon \rho^*$. Define $A_s(v^{\rho^{\mathbb{N}}}) :\equiv \neg B(s @ v)$, which denotes that $s @ v$ is a bad extension of s. Then by $\mathsf{LEP}_{<_{|s|}}$ applied to A_s, where $w <_{|s|} v :\equiv w(|s|) < v(|s|)$ we have

$$\exists r A_s(r) \to \exists v(A_s(v) \wedge \forall w <_{|s|} v \neg A_s(w))$$

from which we obtain (quantifying over s),

$$\forall s, r \exists v(\underbrace{A_s(r) \to A_s(v) \wedge \forall w <_{|s|} v \neg A_s(w)}_{C_{s,r}(v)}).$$

Note that $<$ well-founded clearly implies that $<_{|s|}$ is well-founded, and we could have alternatively used $\mathsf{LEP}_<$ applied to the formula $\tilde{A}_s(x) = \exists v \neg B(s * x \mathbin{@} v)$. Written out fully, $C_{s,r}(v)$ is equivalent to

$$\neg B(s \mathbin{@} r) \to \neg B(s \mathbin{@} v) \wedge \forall w(w(|s|) < v(|s|) \to B(s \mathbin{@} w)), \tag{8.1}$$

and denotes that whenever s has a bad extension r, it also has a minimal bad extension v. Now, to construct a minimal bad sequence from an initial bad sequence $u\colon \rho^{\mathbb{N}}$, define the predicate $\tilde{C}^u$ on $(\rho^{\mathbb{N}})^* \times \rho^{\mathbb{N}}$ by

$$\tilde{C}^u_{\langle t_0,\dots,t_{n-1}\rangle}(v) :\equiv (n = 0 \to C_{\langle\rangle,u}(v)) \wedge (n > 0 \to C_{\langle t_0(0),\dots,t_{n-1}(n-1)\rangle,t_{n-1}}(v)).$$

It is clear that $\forall s, r \exists v C_{s,r}(v) \to \forall t \exists v \tilde{C}^u_t(v)$, and by $\mathsf{DC}_{\mathsf{seq}}$ applied to $\tilde{C}^u$ we have

$$\exists vs^{\mathbb{N}\to\rho^{\mathbb{N}}} \forall n \tilde{C}^u_{[vs](n)}(vs_n)$$

Therefore there exists vs such that, defining $\bar{v} :=_{\rho^{\mathbb{N}}} \lambda n\ .\ vs_n(n)$,

$$\forall n(\neg B([\bar{v}](n) \mathbin{@} vs_{n-1}) \to \neg B([\bar{v}](n) \mathbin{@} vs_n) \wedge \forall w(w(n) < \bar{v}(n) \to B([\bar{v}](n) \mathbin{@} w))).$$

using the convention $vs_{-1} := u$. Finally, we want to verify that $\bar{v}$ is a minimal bad sequence. We use induction to show that we have

$$\neg B(u) \to \forall n \underbrace{[\neg B([\bar{v}](n) \mathbin{@} vs_n) \wedge \forall w(w(n) < \bar{v}(n) \to B([\bar{v}](n) \mathbin{@} w))]}_{D(n)}.$$

We have $\neg B(u) \to D(0)$, and because $[\bar{v}](n) \mathbin{@} vs_n = [\bar{v}](n) * \bar{v}(n) \mathbin{@} vs_n = [\bar{v}](n+1) \mathbin{@} vs_n$ we derive

$$\begin{aligned} D(n) &\to \neg B([\bar{v}](n) \mathbin{@} vs_n) \\ &\to \neg B([\bar{v}](n+1) \mathbin{@} vs_n) \\ &\to D(n+1) \end{aligned}$$

where the last implication follows from $C_{[v](n+1),vs_n}(vs_{n+1})$. Therefore $\neg B(u) \to \forall n D(n)$, or equivalently

$$\neg B(u) \to \forall n \neg B([\bar{v}](n) \mathbin{@} vs_n) \wedge \forall n, w(w(n) < \bar{v}(n) \to B([\bar{v}](n) \mathbin{@} w)).$$

Now, using the fact that $B(w) = \exists k B_0([w](k))$ is open we have

$$\begin{aligned} \forall n \neg B([\bar{v}](n) \mathbin{@} vs_n) &\to \forall n, k \neg B_0([[\bar{v}](n) \mathbin{@} vs_n](k)) \\ &\to \forall n \neg B_0([[\bar{v}](n) \mathbin{@} vs_n](n)) \\ &\to \forall n \neg B_0([\bar{v}](n)) \\ &\to \neg B(\bar{v}) \end{aligned}$$

and therefore

$$\neg B(u) \to \neg \bar{B}(\bar{v}) \wedge \forall n, w(w(n) < \bar{v}(n) \to B([\bar{v}](n) \mathbin{@} w)).$$

from which we obtain

$$\exists u \neg B(u) \to \exists v(\neg B(v) \wedge \forall n, w(w(n) < v(n) \to B([v](n) \mathbin{@} w))).$$

□

8.2 The functional interpretation of MBS

We now extract a realizer for the functional interpretation of MBS. To avoid nested expressions like $[[v](n) \mathbin{@} w](k)$ we use the abbreviation $\bar{B}_0(u,k) :\equiv B_0([u](k))$, and so $B(u) = \exists k \bar{B}_0(u,k)$. With all the quantifiers made explicit, the negative translation of MBS is equivalent (over MP^ω) to

$$(*) \quad \exists u \forall k \neg \bar{B}_0(u,k) \to \neg\neg\exists v(\forall k \neg \bar{B}_0(v,k) \wedge \overbrace{\forall w, n(w(n) < v(n) \to \exists k \bar{B}_0([v](n) \mathbin{@} w, k)}^{C[v]})).$$

The functional interpretation of $C[v]$ is given by

$$|C[v]|^{\gamma}_{n,w} \equiv w(n) < v(n) \to \bar{B}_0([v](n) \mathbin{@} w, \gamma n w)$$

for $\gamma\colon \mathbb{N} \times \rho^{\mathbb{N}} \to \mathbb{N}$, and therefore the conclusion of $(*)$ is equivalent to

$$\neg\neg\exists v, \gamma \forall k, n, w(\neg \bar{B}_0(v,k) \wedge |C[v]|^{\gamma}_{n,w})$$

which is partially interpreted as

$$\forall F, N, W \exists v, \gamma(\neg \bar{B}_0(v, F_{v\gamma}) \wedge |C[v]|^{\gamma}_{N_{v\gamma}, W_{v\gamma}})$$

for $F\colon \kappa \to \mathbb{N}$, $N\colon \kappa \to \mathbb{N}$ and $W\colon \kappa \to \rho^{\mathbb{N}}$, where $\kappa :\equiv \rho^{\mathbb{N}} \times (\mathbb{N} \times \rho^{\mathbb{N}} \to \mathbb{N})$. This means that the functional interpretation of $(*)$ is given by

$$|\mathsf{MBS}^{\mathrm{N}}|^{\Phi,V,\Gamma}_{u,F,N,W} :\equiv \neg \bar{B}_0(u, \Phi_u) \to \neg \bar{B}_0(V_u, F_{V_u \Gamma_u}) \wedge |C[V_u]|^{\Gamma_u}_{N_{V_u\Gamma_u}, W_{V_u\Gamma_u}}$$

for $\Phi\colon \kappa' \to \mathbb{N}$, $V\colon \kappa' \to \rho^{\mathbb{N}}$ and $\Gamma\colon \kappa' \to (\mathbb{N} \times \rho^{\mathbb{N}} \to \mathbb{N})$ where κ' is the type of $\langle u, F, N, W\rangle$. Note that in the preceding formula and throughout we suppress the full dependencies on Φ, V and Γ, so by V_u we mean $V_{u,F,V,\Gamma}$. Written fully, $|\mathsf{MBS}^{\mathrm{N}}|^{\Phi,V,\Gamma}_{u,F,N,W}$ becomes

$$\begin{cases} \neg \bar{B}_0(u, \Phi_u) \to \neg \bar{B}_0(V_u, F_{V_u\Gamma_u}) \wedge (W_{V_u\Gamma_u}(N_{V_u\Gamma_u}) < V_u(N_{V_u\Gamma_u}) \\ \to \bar{B}_0([V_u](N_{V_u\Gamma_u}) \mathbin{@} W_{V_u\Gamma_u}, \Gamma_u N_{V_u\Gamma_u} W_{V_u\Gamma_u})). \end{cases}$$

Roughly speaking, this states that for a proposed bad sequence u there exists an *approximation* V_u to a minimal bad sequence whose minimality is witnessed by Γ_u, where the counterexample F challenges that V_u is approximately bad, while N and W challenge it being approximately minimal, calibrating in a sense the 'length' and 'depth' of the approximation respectively.

Our task is to construct terms Φ, V and Γ parametrised by u, F, V and Γ (and of course implicitly by the characteristic function t_{B_0} for the quantifier-free formula B_0), that satisfy $|\mathsf{MBS}^{\mathrm{N}}|^{\Phi,V,\Gamma}_{u,F,W,N}$.

The functional interpretation of the least element principle

The minimal-bad-sequence argument contains two non-trivial classical principles: dependent choice and the least element principle. We can interpret the former using the product of selection functions as described in Chapter 5, but it remains to determine how we can interpret the latter. We now solve the ND interpretation of $\mathsf{LEP}_<$, recalling that $\mathsf{LEP}_<$ is given by

$$\exists x^\rho A(x) \to \exists y (A(y) \wedge \forall z < y \neg A(z)).$$

To do so we introduce a symbol $\mathsf{R}_{(\rho,<),\sigma}$ for *transfinite recursion* of type σ over $(\rho,<)$ with defining equation

$$\mathsf{R}^g_<(x^\rho) \overset{\sigma}{:=} g_x(\lambda y\ .\ \mathsf{R}^g_<(y) \text{ if } y < x).$$

where as usual ' if $y < x$' is shorthand for if $y < x$, else 0_σ' (note that for the usual ordering $(\mathbb{N},<)$ on natural numbers $\mathsf{R}_{(\mathbb{N},<),\sigma}$ is equivalent to the primitive recursor R_σ). A similar approach is used by Schwichtenberg in [81] where transfinite recursion is used to solve the functional interpretation of the rule of transfinite induction, and we imagine that his realizer is somewhat similar to ours. We restrict ourselves to instances of $\mathsf{LEP}_<$ over Π_1-formulas as this is sufficient for our application of the least element principle.

Remark 8.5. In fact, it is also sufficient for the general case: Suppose that $A(x)$ has ND interpretation $\exists a^{\tilde{\rho}} \forall b^\sigma |A(x)^{\mathrm{N}}|^a_b$. Then the least element principle for A is equivalent to

$$\exists x^\rho, a^{\tilde{\rho}} \forall b^\sigma |A(x)^{\mathrm{N}}|^a_b \to \exists y, a (\forall b |A(y)^{\mathrm{N}}|^a_b \wedge \forall z < y \forall a \exists b \neg |A(z)^{\mathrm{N}}|^a_b)$$

which is just the least element principle for the Π_1-formula $\tilde{A}(x^\rho, a^{\tilde{\rho}}) :\equiv \forall b^\sigma |A(x)^{\mathrm{N}}|^a_b$ on the well-founded order $\langle x, a\rangle \tilde{<} \langle y, a'\rangle :\equiv x < y$.

Let $A(x) \equiv \forall k A_0(x,k)$. We want to realize the negative translation of $\mathsf{LEP}_<$, which is equivalent to

$$\exists x^\rho \forall k A_0(x,k) \to \neg\neg\exists y(\forall k A_0(y,k) \wedge \forall z < y \exists k \neg A_0(z,k)). \tag{8.2}$$

Partially interpreting the conclusion gives us

$$\exists x \forall k A_0(x,k) \to \forall p \exists y, g(A_0(y, p^0 yg) \wedge (p^1 yg < y \to \neg A_0(p^1 yg, g(p^1 yg))))$$

where $p\colon \sigma \to \tau \times \rho$ for $\sigma := \rho \times (\rho \to \tau)$. Finally, we have

$$|\mathsf{LEP}^N_<|^{\varepsilon,K}_{x,p} \equiv \begin{cases} A_0(x, Kxp) \to \\ \underbrace{A_0(\varepsilon^0_x(p), p^0(\varepsilon_x p)) \wedge (p^1(\varepsilon_x p) < \varepsilon^0_x(p) \to \neg A_0(p^1(\varepsilon_x p), \varepsilon^1_x p(p^1(\varepsilon_x p))))}_{D^\varepsilon_{x,p}} \end{cases}.$$

where $\varepsilon\colon \rho \times (\sigma \to \tau \times \rho) \to \sigma$ and $K\colon \rho \times (\sigma \to \tau \times \rho) \to \tau$.

Lemma 8.6. *Given u and p, define $\varepsilon_x\colon J_{\tau\times\rho}\sigma$ via transfinite recursion as*

$$\varepsilon_x(p^{\sigma\to\tau\times\rho}) \overset{\sigma}{:=} \begin{cases} \langle x, f^p_x\rangle & \text{if } \phi(p,x) \\ \varepsilon_{p^1(x,f^p_x)}(p) & \text{otherwise,} \end{cases}$$

where ϕ is the quantifier-free formula defined by

$$\phi(p,x) :\equiv p^1(x, f^p_x) < x \to \neg A_0(p^1(x, f^p_x), f^p_x(p^1(x, f^p_x)))$$

and $f^p_x\colon \rho \to \tau$ is also defined via transfinite recursion as

$$f^p_x := \lambda y \,.\, p^0(y, f^p_y) \text{ if } y < x.$$

Note that ε_x is well-defined since $\neg\phi(p,x) \to p^1(x, f^p_x) < x$. Finally, define $Kxp := p^0(x, f^p_x)$. Then ε and K satisfy $\forall x, p|\mathsf{LEP}^N_<|^{\varepsilon,K}_{x,p}$, or more specifically, the formula $\forall x, pE^{\varepsilon,K}_{x,p}$ where

$$E^{\varepsilon,K}_{x,p} :\equiv \begin{cases} (\phi(p,x) \to |\mathsf{LEP}^N_<|^{\varepsilon,K}_{x,p})\wedge \\ (\neg\phi(p,x) \to D^\varepsilon_{x,p}) \end{cases}$$

(for $D^\varepsilon_{x,p}$ the conclusion of $|\mathsf{LEP}^N_<|^{\varepsilon,K}_{x,p}$ as defined above) which clearly implies $\forall x, p|\mathsf{LEP}^N_<|^{\varepsilon,K}_{x,p}$.

Remark 8.7. We show that our realizer for $\mathsf{LEP}_<$ satisfies a seemingly stronger formula $E^{\varepsilon,K}_{x,p}$. This formula makes explicit the fact that our realizer only uses the premise for $|\mathsf{LEP}^N_<|^{\varepsilon,K}_{x,p}$ in the first case that $\phi(p,x)$ holds, where $\langle x, f^p_x\rangle$ already suffices to solve it.

This property will become useful when it comes to verifying inductively that our main realizer, which is a product of selection functions $\varepsilon_x p$ corresponding to instances of $\mathsf{LEP}_<$, forms a minimal-bad-sequence - as we will require the conclusion of one instance of $|\mathsf{LEP}^N_<|$ to form the premise of the next. This 'short-cut' allows us to suppress additional formalism that would be present in a rigid extraction from a fully formal proof.

Proof of Lemma 8.6. For arbitrary fixed p we use (quantifier-free) transfinite induction on $E^{\varepsilon,K}_{x,p}$ to prove $\forall x E^{\varepsilon,K}_{x,p}$. All we need to do is show that $E^{\varepsilon,K}_{x,p}$ is progressive i.e. that for each x we have

$$\forall y < x E^{\varepsilon,K}_{y,p} \to E^{\varepsilon,K}_{x,p}.$$

We must deal with two cases.

(a) $\phi(p,x)$ holds, in which case $\varepsilon_x p = \langle x, f^p_x \rangle$. Then we have

$$A_0(x, p^0(x, f^p_x)) \to A_0(x, p^0(x, f^p_x)) \wedge \phi(p,x)$$

which is equivalent to $|\mathsf{LEP}^N_<|^{\varepsilon,K}_{x,p}$.

(b) $\phi(p,x)$ fails, and therefore $p^1(x,f^p_x) < x \wedge A_0(p^1(x,f^p_x), f^p_x(p^1(x,f^p_x)))$ and $\varepsilon_x p = \varepsilon_y p$ for $y = p^1(x,f^p_x) < x$. Then by hypothesis we have $E^{\varepsilon,K}_{y,p}$ and hence $|\mathsf{LEP}^N_<|^{\varepsilon,K}_{y,p}$, which is

$$A_0(y, p^0(y, f^p_y)) \to D^\varepsilon_{y,p}. \tag{8.3}$$

But $A_0(p^1(x,f^p_x), f_x(p^1(x,f^p_x)))$ is equivalent to $A_0(y, p^0(y,f^p_y))$ since $f_x(p^1(x,f^p_x)) = f_x(y) = p^0(y,f^p_y)$. This is the premise of (8.3), so $D^\varepsilon_{y,p}$, and therefore $D^\varepsilon_{x,p}$ holds since $\varepsilon_x p = \varepsilon_y p$ implies $D^\varepsilon_{x,p} \leftrightarrow D^\varepsilon_{y,p}$. □

Realizing $\forall u, F, W, N |\mathsf{MBS}^{\mathrm{N}}|^{\Phi,V,\Gamma}_{u,F,W,N}$

We are now ready to extract a realizer from Nash-Williams' proof of the minimal-bad-sequence argument. Each of the following results correspond intuitively to one of the main steps in our formalisation of the proof. Our first step is to interpret the instance of the least element principle used in the classical construction of the minimal bad sequence (8.1).

Lemma 8.8. *Define the selection function $\varepsilon_{s,v}\colon J_{\mathbb{N}\times\rho^{\mathbb{N}}}\sigma$, where $\sigma := \rho^{\mathbb{N}} \times (\rho^{\mathbb{N}} \to \mathbb{N})$ via transfinite recursion on $<_{|s|}$, where as before $v' <_{|s|} v :\equiv v'(|s|) < v(|s|)$, as*

$$\varepsilon_{s,v}(p^{\sigma\to\mathbb{N}\times\rho^{\mathbb{N}}}) \overset{\sigma}{:=} \begin{cases} \langle v, f^p_{|s|,v} \rangle & \text{if } \phi(s,v,p) \\ \varepsilon_{s,p^1(v,f^p_{|s|,v})}(p) & \text{otherwise} \end{cases}$$

where

$$\phi(s,v,p) :\equiv p^1(v, f^p_{|s|,v})(|s|) < v(|s|) \to \bar{B}_0(s \mathbin{@} p^1(v, f^p_{|s|,v}), f^p_{|s|,v}(p^1(v, f^p_{|s|,v}))))$$

and $f^p_{|s|,v}$ is defined using transfinite recursion on $<_{|s|}$ as

$$f^p_{|s|,v} := \lambda w \,.\, p^0(w, f^p_{|s|,w}) \text{ if } w(|s|) < v(|s|).$$

Define $K_{s,v}p := p^0(v, f^p_{|s|,v})$. Then we have $\forall s,v,p E^{\varepsilon,K}_{s,v,p}$ where $E^{\varepsilon,K}_{s,v,p}$ is defined as in Lemma 8.6 by

$$E^{\varepsilon,K}_{s,v,p} :\equiv \left\{ \begin{array}{l} (\phi(s,v,p) \to (\neg\bar{B}_0(s \mathbin{@} v, K_{s,v}p) \to D^\varepsilon_{s,v,p}))\wedge \\ (\neg\phi(s,v,p) \to D^\varepsilon_{s,v,p}) \end{array} \right.$$

for

$$\underbrace{\neg\bar{B}_0(s \mathbin{@} \varepsilon^0_{s,v}p, p^0(\varepsilon_{s,v}p)) \wedge (p^1(\varepsilon_{s,v}p)(|s|) < \varepsilon^0_{s,v}p(|s|) \to \bar{B}_0(s \mathbin{@} p^1(\varepsilon_{s,v}p), \varepsilon^1_{s,v}p(p^1(\varepsilon_{s,v}p))))}_{D^{\varepsilon}_{s,v,p}}$$

Proof. This is entirely analogous to the proof of Lemma 8.6, setting $<:\equiv<_{|s|}$ and $A_0(v,k) :\equiv \neg\bar{B}_0(s \mathbin{@} v, k)$. As before, we note that $\mathsf{TI}_{<_{|s|}}$ follows from $\mathsf{TI}_{<}$, and similarly $\mathsf{R}_{<_{|s|}}$ is easily definable from $\mathsf{R}_{<}$. □

The formula $E^{\varepsilon,K}_{s,v,p}$ is a computational interpretation of the statement that if s has a bad extension v, then it has a bad extension that is minimal with respect to $<_{|s|}$. The formula $D^{\varepsilon}_{s,v,p}$ says that the term $\varepsilon^0_{s,v}p$ represents an approximation to this minimal extension with $\varepsilon^1_{s,v}p$ witnessing its minimality, all relative to the counterexample p. The term $K_{s,v}p$ indicates how we use the assumption that $s \mathbin{@} v$ is bad to construct the approximation - in particular this assumption is not required at all if $\neg\phi(s,v,p)$ holds, which is the case if we find an approximate bad extension to s less than v (cf. Remark 8.7).

Now, we use EPS to interpret the instance of $\mathsf{DC}_{\mathsf{seq}}$ used to construct the minimal-bad-sequence itself. Our selection functions are capable of constructing minimal bad extensions at each point, so their unbounded product gives us an approximation to a bad sequence approximately minimal over all points.

Lemma 8.9. *Define $\tilde{\varepsilon}^u \colon \sigma^* \to J_{\mathbb{N}\times\rho^{\mathbb{N}}}\sigma$ parametrised by $u \colon \rho^{\mathbb{N}}$ as*

$$\tilde{\varepsilon}^u_{T^{\sigma^*}}(p^{\sigma\to\mathbb{N}\times\rho^{\mathbb{N}}}) \overset{\sigma}{:=} \begin{cases} \varepsilon_{\langle\rangle,u}(p) & \text{if } |T| = 0 \\ \varepsilon_{\langle\pi_0 T_0(0),\dots,\pi_0 T_{n-1}(n-1)\rangle,\pi_0 T_{n-1}}(p) & \text{if } |T| > 0. \end{cases}$$

where the type σ and the selection function $\varepsilon_{s,v}$ are defined as in Lemma 8.6. Note that $\pi_0 T$ is analogous to the variable t in the formal proof, while $\pi_1 T$ represents a sequence of functionals witnessing minimality that are hidden in the formal proof. Now given $q \colon \sigma^{\mathbb{N}} \to \mathbb{N}\times\rho^{\mathbb{N}}$ and $\varphi \colon \sigma^{\mathbb{N}} \to \mathbb{N}$ define

$$vs^u \overset{(\rho^{\mathbb{N}})^{\mathbb{N}}}{:=} \pi_0\mathsf{EPS}^{\tilde{\varepsilon}^u,q,\varphi}(\langle\rangle)$$
$$fs^u \overset{(\rho^{\mathbb{N}}\to\mathbb{N})^{\mathbb{N}}}{:=} \pi_1\mathsf{EPS}^{\tilde{\varepsilon}^u,q,\varphi}(\langle\rangle).$$

Then by Theorem 5.7, using the abbreviation $v^u :\equiv \lambda n.vs^u_n(n)$ and the convention $vs^u_{-1} :\equiv u$, and setting $p_n = p_{[vs^u_n,fs^u_n](n)}$ as defined as in Theorem 5.7, for all $n \leq \varphi(vs^u, fs^u)$ we have

$$\langle vs^u_n, fs^u_n\rangle = \tilde{\varepsilon}^u_{[vs^u,fs^u](n)}(p_n) = \langle \varepsilon^0_{[v^u](n),vs^u_{n-1}}(p_n), \varepsilon^1_{[v^u](n),vs^u_{n-1}}(p_n)\rangle$$
$$p_n(\varepsilon_{[v^u](n),vs^u_{n-1}}(p_n)) = q(vs^u, fs^u)$$

This implies that for all $n \leq \varphi(vs^u, fs^u)$, vs^u and fs^u satisfy (suppressing the parameter u)

$$\begin{cases} \phi([v](n), vs_{n-1}, p_n) \to (\neg\bar{B}_0([v](n) \mathbin{@} vs_{n-1}, K_{[v](n),vs_{n-1}}p_n) \to \tilde{D}^{vs,fs}_{n,q}) \\ \neg\phi([v](n), vs_{n-1}, p_n) \to \tilde{D}^{vs,fs}_{n,q} \end{cases}$$

where K is defined as in Lemma 8.6 and

$$\tilde{D}^{vs,fs}_{n,q} :\equiv \begin{cases} \neg\bar{B}_0([v](n) \mathbin{@} vs_n, q^0(vs,fs)) \wedge \\ (q^1(vs,fs)(n) < v(n) \to \bar{B}_0([v](n) \mathbin{@} q^1(vs,fs), fs_n(q^1(vs,fs)))). \end{cases}$$

Proof. The lemma follows by substituting, for each $n \leq \varphi(vs^u, fs^u)$, $s = [v^u](n)$, $v = v_{n-1}$ and $p = p_n$ into Lemma 8.8. We have

$$\tilde{D}^{vs^u,fs^u}_{n,q} \leftrightarrow D^{\varepsilon}_{[v^u](n),vs_{n-1},p_n}$$

by Spector's equations. □

Now we verify by induction that from our initial assumption that if u is bad the diagonal sequence v is minimal bad. The next lemma is the constructive analogue of the statement that $\neg B(u) \to \forall n D(n)$ in the formal proof.

Lemma 8.10. *Given u, q and φ let everything be defined as in Lemma 8.9. Then*

$$\neg\bar{B}_0(u, q^0(vs^u, fs^u)) \to \forall n \leq \varphi(vs^u, fs^u)\tilde{D}^{vs^u,fs^u}_{n,q}.$$

Proof. This is done by Lemma 8.9 and induction on $\tilde{D}^{vs^u,fs^u}_{n,q}$. First we show that (suppressing the parameter u in vs, fs)

$$\neg\bar{B}_0(u, q^0(vs,fs)) \to \tilde{D}^{vs,fs}_{0,q}.$$

This follows from Lemma 8.9 for $n = 0$. Assume that $\phi(\langle\rangle, u, p_0)$ (if not then we get $\tilde{D}^{vs,fs}_{0,q}$ anyway). Then by definition $\varepsilon_{\langle\rangle,u}(p_0) = \langle u, f^{p_0}_{0,u}\rangle$, which implies that $K_{\langle\rangle,u}p_0 = p^0_0(\varepsilon_{\langle\rangle,u}(p_0)) = q^0(vs,fs)$ and so

$$\neg\bar{B}_0(u, q^0(vs,fs)) \to \neg\bar{B}_0(u, K_{\langle\rangle,u}p_0) \to \tilde{D}^{vs,fs}_{0,q}.$$

Now for $n < \varphi(vs,fs)$ we show that

$$\tilde{D}^{vs,fs}_{n,q} \to \tilde{D}^{vs,fs}_{n+1,q}.$$

First, since $[v](n) \mathbin{@} vs_n = [v](n+1) \mathbin{@} vs_n$ we have

$$\tilde{D}^{vs,fs}_{n,q} \to \neg\bar{B}_0([v](n) \mathbin{@} vs_n, q^0(vs,fs)) \to \neg\bar{B}_0([v](n+1) \mathbin{@} vs_n, q^0(vs,fs)).$$

Now, by Lemma 8.9, assuming that $\phi([v](n+1), vs_n, p_{n+1})$ (otherwise we get $\tilde{D}^{vs,fs}_{n+1,q}$ anyway) we have $\varepsilon_{[v](n+1),vs_n}(p_{n+1}) = \langle vs_n, f^{p_{n+1}}_{n+1,vs_n}\rangle$ and therefore $K_{[v](n+1),vs_n}p_{n+1} = p^0_{n+1}(\varepsilon_{[v](n+1),vs_n}(p_{n+1})) = q^0(vs, fs)$ and so

$$\neg\bar{B}_0([v](n+1) \mathbin{@} vs_n, q^0(vs, fs)) \to \neg\bar{B}_0([v](n+1) \mathbin{@} vs_n, K_{[v](n+1),vs_n}p_{n+1}) \to \tilde{D}^{vs,fs}_{n+1,q}.$$

Therefore by induction from $n = 0$ to $n = \varphi(vs^u, fs^u) - 1$ we get $\tilde{D}^{vs,fs}_{0,q} \to \forall n \leq \varphi(vs^u, fs^u)\tilde{D}^{vs,fs}_{n,q}$, and the result follows. □

Finally, we define q and φ in terms of the counterexample functionals F, W and N in order to complete the construction.

Theorem 8.11. *Suppose we have an arbitrary sequence $u\colon \rho^{\mathbb{N}}$ and functionals $F\colon \kappa \to \mathbb{N}$, $N\colon \kappa \to \mathbb{N}$ and $W\colon \kappa \to \rho^{\mathbb{N}}$ where $\kappa\colon \rho^{\mathbb{N}} \times (\mathbb{N} \times \rho^{\mathbb{N}} \to \mathbb{N})$. Define $q^{F,W}$, and $\varphi_{F,N}$ on $\langle vs, fs\rangle\colon (\rho^{\mathbb{N}})^{\mathbb{N}} \times (\rho^{\mathbb{N}} \to \mathbb{N})^{\mathbb{N}} \cong \sigma^{\mathbb{N}}$ by*

$$\begin{aligned} q^{F,W}(vs, fs) &\overset{\mathbb{N}\times\rho^{\mathbb{N}}}{:=} \langle F(v, fs), W(v, fs)\rangle \\ \varphi^{F,N}(vs, fs) &\overset{\mathbb{N}}{:=} \max\{F(v, fs), N(v, fs)\} \end{aligned}$$

where $v := \lambda n \,.\, vs_n(n)$. Finally, define $vs^u \overset{(\rho^{\mathbb{N}})^{\mathbb{N}}}{:=} \pi_0\mathsf{EPS}^{\tilde{\varepsilon}^u, q^{F,W}, \varphi^{F,N}}(\langle\rangle)$, $fs^u \overset{(\rho^{\mathbb{N}}\to\mathbb{N})^{\mathbb{N}}}{=} \pi_1\mathsf{EPS}^{\tilde{\varepsilon}^u, q^{F,W}, \varphi^{F,N}}(\langle\rangle)$ and $v^u := \lambda n \,.\, vs^u_n(n)$, for $\tilde{\varepsilon}$ defined as in Lemma 8.9. Then

$$\begin{aligned} \Phi_{F,N,W_u} &:= F(v^u, fs^u) \\ \Gamma_{F,N,W,u} &:= fs^u \\ V_{F,N,W,u} &:= v^u \end{aligned}$$

satisfy $|\mathsf{MBS}^{\mathrm{N}}|^{\Phi,V,\Gamma}_{u,F,W,N}$.

Proof. By Lemma 8.10 we have (suppressing dependencies F, N and W)

$$\neg B_0(u, \Phi_u) \to \forall n \leq \max\{F_{V_u\Gamma_u}, N_{V_u\Gamma_u}\}\tilde{D}^{vs^u,\Gamma_u}_{n,q^{F,W}}.$$

Firstly, setting $n = F_{V_u\Gamma_u}$ yields

$$\neg\bar{B}_0(u, \Phi_u) \to \tilde{D}^{vs^u,\Gamma_u}_{F_{V_u\Gamma_u},q^{F,W}} \to \neg\bar{B}_0([V_u](F_{V_u\Gamma_u}) \mathbin{@} vs^u_{F_{V_u\Gamma_u}}, F_{V_u\Gamma_u}) \to \neg\bar{B}_0(V_u, F_{V_u\Gamma_u})$$

where for the last implication we use $[[V_u](n) \mathbin{@} v](n) = [V_u](n)$. Secondly, setting $n = N_{V_u\Gamma_u}$ yields

$$\begin{aligned} &\neg\bar{B}_0(u, \Phi_u) \to \tilde{D}^{vs^u,\Gamma_u}_{N_{V_u\Gamma_u},q^{F,W}} \\ &\qquad \to W_{V_u\Gamma_u}(N_{V_u\Gamma_u}) < V_u(N_{V_u\Gamma_u}) \to \bar{B}_0([V_u](N_{V_u\Gamma_u}) \mathbin{@} W_{V_u\Gamma_u}, \Gamma_u(N_{V_u\Gamma_u}, W_{V_u\Gamma_u})). \end{aligned}$$

Combining the two yields

$$\begin{cases} \neg\bar{B}_0(u, \Phi_u) \to \neg\bar{B}_0(V_u, F_{V_u\Gamma_u}) \wedge (W_{V_u\Gamma_u}(N_{V_u\Gamma_u}) < V_u(N_{V_u\Gamma_u}) \\ \to \bar{B}_0([V_u](N_{V_u\Gamma_u}) \mathbin{@} W_{V_u\Gamma_u}, \Gamma_u(N_{V_u\Gamma_u}, W_{V_u\Gamma_u}))) \end{cases}$$

which is $|\mathsf{MBS}^{\mathrm{N}}|^{\Phi,V,\Gamma}_{u,F,W,N}$. □

An analysis of our bar recursive realizer

Having carried out the formal construction of our realizer, we now aim to describe in more intuitive terms how it behaves, and to demonstrate that it forms a clear computational analogue to the minimal-bad-sequence construction of Nash-Williams. As before we can use the semantics of EPS to identify various parts of the realizer with game-theoretic concepts, associating our computational interpretation of MBS with a single partially defined sequential game $\mathcal{G}[u, F, N, W]$.

The functional interpretation of MBS asks for an approximation of the minimal bad sequence construction. Given an initial bad sequence u we must construct a bad sequence V_u and a functional Γ_u witnessing its minimality relative to counterexample functionals F, N and W, and must also produce a point Φ_u such that all this can be verified from $\neg B_0([u](\Phi_u))$.

The basic idea. We build V_u and Γ_u as optimal plays in the sequential game whose selection functions are given by $\tilde{\varepsilon}^u$. Moves in this game consist of a sequence $v\colon \rho^{\mathbb{N}}$ and a functional $f\colon \rho^{\mathbb{N}} \to \mathbb{N}$. The idea is that for an optimal play $\langle vs^u, fs^u\rangle$ of type $(\rho^{\mathbb{N}})^{\mathbb{N}} \times (\rho^{\mathbb{N}} \to \mathbb{N})^{\mathbb{N}}$, setting $v^u := \lambda n.vs^u_n(n)$ the sequence vs^u_n represents a minimal bad extension of $[v^u](n)$ whose minimality is witnessed by $\lambda w.fs^u_n w$, and therefore $V_u = v^u$ represents a minimal-bad-sequence whose minimality as a whole is witnessed by $\Gamma_u = \lambda n, w.fs^u_n w$.

The role of the selection functions is to interpret pointwise instances of $\mathsf{LEP}_<$ in the minimal-bad-sequence construction and produce approximations to minimal bad extensions of $[V_u](n)$ in the construction relative to 'local' counterexample functions p. The product of selection functions enables us to construct a whole sequence V_u that is minimal bad relative to the global counterexample functions q and φ, which are in turn defined in terms of the F, N and W so that V_u is an approximation of sufficient quality to ensure that it is 'bad' at point $F_{V_u\Gamma_u}$ and approximately minimal at point $N_{V_u\Gamma_u}$ relative to $W_{V_u\Gamma_u}$.

classical proof	sequential game $\mathcal{G}[u, F, N, W]$
least element principle	selection functions $\tilde{\varepsilon}^u$
minimal bad sequence	unbounded product $\mathsf{EPS}^{\tilde{\varepsilon}^u, q^{F,W}, \varphi^{F,N}}$

Table 8.1: Functional interpretation of MBS

How the realizer works. The behaviour of our realizer mimics the classical construction of the minimal bad sequence. It is based on a main instance of bar recursion which interprets the overall construction, each step in the bar recursion involving an instance of transfinite recursion implemented by the selection functions to interpret the least element principle.

Given that we have computed a proposed initial segment $\langle vs^u_0, fs^u_0\rangle, \ldots, \langle vs^u_{n-1}, fs^u_{n-1}\rangle$

for our minimal-bad-sequence, where vs^u_{n-1} is an approximately bad extension of $[v^u](n)$ relative to F, the selection function $\varepsilon_{[v^u](n),vs^u_{n-1}}$ picks an extension v of $[v^u](n)$ and a functional f satisfying

$$\neg\bar{B}_0([v^u](n) \mathbin{@} v, F_{\alpha_{v,f}}) \wedge (W_{\alpha_{v,f}}(n) < v(n) \to \bar{B}_0([v^u](n) \mathbin{@} W_{\alpha_{v,f}}, f(W_{\alpha_{v,f}}))$$

where $\alpha_{v,f}$ is obtained from the optimal extension that results from choosing $\langle v, f\rangle$ at round n. Then $[\bar{v}^u](n) * v(n)$ extends the our approximation by one element, and f witnesses its minimality at point n relative to W. The selection function $\varepsilon_{[v^u](n),v^u_{n-1}}$ computes v and f by first trying as default $\langle v, f\rangle = \langle vs^u_{n-1}, f^{p_n}_{n,vs^u_{n-1}}\rangle$, depending on whether or not $\phi([\bar{v}^u](n), vs^u_{n-1}, p_n)$ holds. Testing this condition corresponds to testing whether or not (approximate) minimality of $[v^u](n) * vs^u_{n-1}(n)$ is satisfied. If it is, the computation terminates, with the truth of $\neg\bar{B}_0([v^u](n) \mathbin{@} v, F_{\alpha_{v,f}})$ relying on the assumption that vs_{n-1} is approximately bad relative to the global counterexample function F. Otherwise the selection function changes its mind and chooses $\langle v, f\rangle = \langle t, f^{p_n}_{n,t}\rangle$ where $t := W(\alpha_{vs^u_{n-1}, f^{p_n}_{n,v^u_{n-1}}})$. This time, if $\langle t, f^{p_n}_{n,t}\rangle$ is chosen then $\neg\bar{B}_0([v^u](n) \mathbin{@} v, F_{\alpha_{v,f}})$ is guaranteed by the failure of minimality of vs^u_{n-1}, since this implies $\neg\bar{B}_0([v^u](n) \mathbin{@} t, f^{p_n}_{n,vs^u_{n-1}}(t))$ and we have $f^{p_n}_{n,vs^u_{n-1}}(t) = p^0_n(t, f^{p_n}_{n,t}) = F_{\alpha_{v,f}}$. The selection function now tests minimality of $\langle t, f^{p_n}_{n,t}\rangle$, and if this fails the whole procedure repeats until a suitable minimal extension of $[v^u](n)$ is finally found. Termination of the search is guaranteed by well-foundedness of $<$.

The product of selection functions implements a backtracking procedure over these selection functions that constructs an approximation to the whole minimal-bad-sequence, comprising a constructive analogue of the inductive procedure carried out in the classical proof. On a global level we claim the functionals $f^{p_n}_{n,v}$ are essentially devices that make recursive calls over the lexicographic ordering, and therefore one could potentially characterise our bar recursive realizer as an algorithm that implements some kind of open recursion (see Section 8.3 below), although making this intuition precise would involve a much more detailed analysis of the realizer.

The functional Φ. One important component of our realizer that we have ignored so far is the functional Φ, which interprets the assumption that the initial sequence u is bad and is crucial to enabling us to carry out the *first* step in the computation. In Lemma 8.10 and Theorem 8.11 we show that if $\Phi(u) = F_{V_u,\Gamma_u}$ then $\neg B_0(u, \Phi_u)$ guarantees that our minimal bad sequence V_u is satisfies $\neg B_0(V_u, F_{V_u,\Gamma_u})$. However, actually we only require that $\neg B_0(u, \Phi_u)$ holds in the case that $\phi(n, vs^u_{n-1}, p_n)$ is true for sufficiently many steps in the computation. The failure of this condition is the constructive analogue of discovering a new bad sequence lexicographically less than u, which means that we no longer require the assumption that u is bad. While our construction of V_u and Γ_u seems quite natural, our definition of Φ_u is in fact rather crude, and it would be instructive to give a more intelligent reformulation of Φ_u based on a more careful logical analysis of the classical proof. However, in this dissertation we will not go into any more details on potential computational inefficiencies in our realizers. For now we are satisfied to have obtained an intuitive constructive analogue of Nash-Williams' proof of the minimal-bad-sequence argument in terms of the product of selection functions.

8.3 Open induction and open recursion

As we pointed out in Section 8.1, the minimal-bad-sequence argument is classically equivalent to the principle of *open induction*, which for our restricted form of open predicate is written fully as

$$(*)\quad \forall v\overbrace{(\underbrace{\forall w, n(w(n) < v(n) \to \exists k \bar{B}_0([v](n) \mathbin{@} w, k))}_{C[v]} \to \exists k \bar{B}_0(v, k))}^{D} \to \forall u \exists k \bar{B}_0(u, k).$$

From an intuitionistic point of view, the two principles behave quite differently. The computational content of open induction is explored by Berger in [10], who demonstrates that unlike dependent choice or MBS, open induction is closed under the negative and A-translations, and can be given a direct realizability interpretation using the schema of *open recursion* defined by:

$$\mathsf{OR}^F(u) \overset{\mathbb{N}}{:=} F_u(\lambda n, v\,.\,\mathsf{OR}^F([u](n) \mathbin{@} v) \text{ if } v(n) < u(n)).$$

This offers a simple and elegant alternative to program extraction from classical proofs involving the minimal-bad-sequence argument without having to go through the usual bar recursive interpretation of dependent choice. We quickly sketch Berger's idea, although the reader is encouraged to consult [10] for details. The formula $C[v]$ (defined in $(*)$ above) has modified realizability interpretation

$$\begin{aligned}\gamma\ mr\ C[v] &\equiv \forall n, w(w(n) < v(n) \to \bar{B}_0([v](n) \mathbin{@} w, \gamma n w)) \\ &\leftrightarrow \forall n, w |C[v]|^{\gamma}_{n,w},\end{aligned}$$

where $\gamma \colon \mathbb{N} \times \rho^{\mathbb{N}} \to \mathbb{N}$, and therefore the subformula D (cf. $(*)$) is interpreted as

$$F\ mr\ D \equiv \forall v, \gamma(\gamma\ mr\ C[v] \to \bar{B}_0(v, Fv\gamma))$$

where $F \colon \rho^{\mathbb{N}} \times (\mathbb{N} \times \rho^{\mathbb{N}} \to \mathbb{N}) \to \mathbb{N}$. Finally, $\Phi\ mr\ \mathsf{OI} \equiv \forall F(F\ mr\ D \to \forall u \bar{B}_0(u, \Phi^F u))$, or alternatively:

$$\Phi\ mr\ \mathsf{OI} \equiv \forall F(\forall v, \gamma(\forall n, w|C[v]|^{\gamma}_{n,w} \to \bar{B}_0(v, Fv\gamma)) \to \forall u \bar{B}_0(u, \Phi^F u)).$$

where $\Phi \colon (\rho^{\mathbb{N}} \times (\mathbb{N} \times \rho^{\mathbb{N}} \to \mathbb{N}) \to \mathbb{N}) \times \rho^{\mathbb{N}} \to \mathbb{N}$.

Theorem 8.12 (Berger [10]). $\mathsf{E\text{-}HA}^{\omega} + \mathsf{Cont} + \mathsf{OI} + \mathsf{OR} \vdash \Phi\ mr\ \mathsf{OI}$ *where* Φ *is a closed term of* $\mathsf{E\text{-}HA}^{\omega} + \mathsf{OR}$.

Proof. In [10] this theorem holds generally for open predicates of the form $\forall n C([u](n)) \to \exists n B([u](n))$ for C arbitrary and B a Σ-formula ($\forall$, $\to$ free and only containing predicates ranging over variables of type level 0). In our restricted case of Σ^0_1-open formulas, $\Phi := \mathsf{OR}$ does the trick. To see this, suppose that $F \; mr \; D$ - we must show that $\forall u \bar{B}_0(u, \mathsf{OR}^F(u))$. Now assuming continuity, the truth of predicate $P(u) :\equiv \bar{B}_0(u, \mathsf{OR}^F(u))$ depends only on a initial segment of u, which means that P is equivalent to an open predicate, so we can use open induction on $P(u)$. Given v, suppose by the induction hypothesis that $P(w)$ holds for all $w <_{\text{lex}} v$ i.e. $\forall n, w(w(n) < v(n) \to \bar{B}_0([v](n) \mathbin{@} w, \mathsf{OR}^F([v](n) \mathbin{@} w)))$. Then

$$\gamma = \lambda n, w \; . \; \mathsf{OR}^F([v](n) \mathbin{@} w) \text{ if } w(n) < v(n)$$

realizes $C[v]$, and therefore since $F \; mr \; D$ we have $\bar{B}_0(v, Fv\gamma) \equiv \bar{B}_0(v, \mathsf{OR}^F(v))$ which is $P(v)$. This verifies progressiveness of P, and so we have $\forall u P(u)$ and we're done. □

'Explicitly controlled' open recursion

It is not difficult to show that open induction is equivalent to its negative translation over $\mathsf{QF\text{-}AC} + \mathsf{MP}^\omega$, and therefore by the soundness theorems for the ND interpretation and the fact that open induction is classically provable from dependent choice a direct realizer for its functional interpretation can be found[1]. In fact, the functional interpretation of OI coincides with the ND interpretation of its contrapositive MBS, so our bar recursive realizer for MBS also realizes OI. However, it is natural to ask whether an analogue to Theorem 8.12 can be found, whereby a direct realizer for the functional interpretation of open induction can be given in terms of open recursion as opposed to bar recursion. For the functional interpretation, in light of Spector's soundness proof we would expect to be able to construct a form of open recursion that is 'explicitly' well-founded using Spector's condition as opposed to implicitly relying on continuity, and would therefore be computationally weaker than full open recursion in the same way that Spector's bar recursion is weaker than modified bar recursion (see Chapter 9).

This short section contains some preliminary results and ideas in the spirit of [10] that explore the idea of whether an alternative to Spector's bar recursion based on open recursion can be devised to extract witnesses for the functional interpretation of proofs involving open induction or the minimal-bad-sequence argument. What follows should be considered nothing more than a starting point for future research, and many details below are merely sketched or left as conjectures. Our aim is simply to design, in an informal manner, a procedure that directly realizes the functional interpretation of open induction and perhaps reflects the computational content of this principle more naturally than our bar recursive realizer would.

[1] Note that the prima facie existence of a bar recursive realizer for OI is already implicitly established in [10, 20] in which it is shown that OI follows from bar induction. Similarly, in [44, 92] it is shown directly that open induction on the Cantor space follows from the double negation shift and Markov's principle.

Let us first establish some notation. For a functional $\Phi\colon \rho^{\mathbb{N}} \to \mathbb{N}$ and a sequence $\rho^{\mathbb{N}}$, define $\Phi\{u\}\colon \rho^{\mathbb{N}} \times (\rho^{\mathbb{N}} \to \mathbb{N})^{\mathbb{N}}$ by

$$\Phi\{u\} := \langle u; \lambda n, w \;.\; \Phi([u](n) \mathbin{@} w) \text{ if } w(n) < u(n)\rangle$$

so $\Phi\{u\}$ consists of the sequence u along with the functional Φ restricted to arguments lexicographically less than u. In this way we can concisely write the defining equation of open recursion as

$$\mathsf{OR}^F(u) = F(\mathsf{OR}^F\{u\}).$$

Now, treating $\Phi\{u\}$ as a sequence of type $\sigma^{\mathbb{N}}$ for $\sigma := \rho \times (\rho^{\mathbb{N}} \to \mathbb{N})$, we have

$$\overline{\Phi\{u\}, M} \overset{\sigma^{\mathbb{N}}}{=} \langle \overline{u, M}; \lambda n < M, w \;.\; \Phi([u](n) \mathbin{@} w) \text{ if } w(n) < u(n)\rangle.$$

where $(\lambda n < M.fn)(m) = 0$ for $m \geq M$. Given a functional $\varphi\colon \sigma^{\mathbb{N}} \to \mathbb{N}$ let

$$\Phi\{u\}_\varphi \overset{\sigma^{\mathbb{N}}}{:=} \overline{\Phi\{u\}, \mu_{\mathrm{Sp}}(\Phi\{u\}, \varphi)}$$

where $M := \mu_{\mathrm{Sp}}(\Phi\{u\}, \varphi)$ is the least number satisfying $\varphi(\overline{\Phi\{u\}, M}) < M$. Note that $\Phi\{u\}_\varphi$ is primitive recursively *definable* in Φ: for arbitrary $\alpha\colon \sigma^{\mathbb{N}}$ we can write

$$\alpha_\varphi(i) := \begin{cases} 0_\sigma & \text{if } \exists k \leq i(\varphi(\overline{\alpha, k}) < k) \\ \alpha(i) & \text{otherwise} \end{cases}$$

and we obtain $\alpha_\varphi = \overline{\alpha, \mu_{\mathrm{Sp}}(\varphi, \alpha)}$ provably in Spec, which we need to confirm the existence of $\mu_{\mathrm{Sp}}(\varphi, \alpha)$ (cf. Proposition 4.2). Let us now define a variant of open recursion EOR that is *explicitly* controlled by the functional φ:

$$\mathsf{EOR}^{F,\varphi}(u) = F(\mathsf{EOR}^{F,\varphi}\{u\}_\varphi).$$

While open recursion OR in the sense of Berger is implicitly well-founded by open induction and continuity of F (see Chapter 9 or [10] for the proof that OR is total), EOR is explicitly well-founded because we specify, via φ, the finite portion of u we wish to use and therefore totality of $\mathsf{EOR}^{F,\varphi}$ is by definition an open property on u.

Remark 8.13. While the functional EOR is definable from OR, we *conjecture* that EOR is in fact of the same computational strength as Spector's bar recursion, whereas as we show in Chapter 12, full open recursion is equivalent to the stronger modified bar recursion.

A partial functional interpretation of OI, in which we do not extract realizers for v and γ, is given by

$$(\dagger) \quad \exists \Phi \forall u, F, N, W (\forall v, \gamma (|C[v]|^{\gamma}_{N_{v\gamma}, W_{v\gamma}} \to \bar{B}_0(v, Fv\gamma)) \to \bar{B}_0(u, \Phi(u))$$

which of course differs from the modified realizability interpretation in that the quantifiers $\forall n, w$, not permitted by the functional interpretation, are eliminated in favour of counterexample functionals N and W. We show that (†) can be realized using EOR.

Proposition 8.14. *Define* $\varphi^{N,F} \colon \sigma^{\mathbb{N}} \to \mathbb{N}$ *by* $\varphi^{N,F} v\gamma := \max\{F_{v\gamma}, N_{v\gamma}\}$. *Then* $\Phi^{F,N}(u) := \mathsf{EOR}^{\varphi,F}(u)$ *realizes* (†), *provably in* $\mathsf{Spec} + \mathsf{OI}$.

Proof. We suppress the parameters on $\varphi^{N,F}$. Fix F, N and W and assume that

$$(+) \quad \forall v, \gamma(|C[v]|^{\gamma}_{N_{v\gamma}, W_{v\gamma}} \to \bar{B}_0(v, Fv\gamma))$$

holds. We use open induction on the open predicate $P(u) := \bar{B}_0(u, \mathsf{EOR}^{F,\varphi}(u))$ (P is open because by definition EOR only looks at a finite initial segment of u). To establish progressiveness, let's assume that $P(w)$ holds for all w lexicographically less than u. Now, define

$$\begin{aligned} \langle v, \gamma \rangle &:= \mathsf{EOR}^{F,\varphi}\{u\}_{\varphi} \\ &= \langle \overline{u, M}; \lambda n < M, w \,.\, \mathsf{EOR}^{F,\varphi}([u](n) @ w) \text{ if } w(n) < u(n) \rangle \end{aligned}$$

where (assuming Spec) $M = \mu_{\mathrm{Sp}}(\mathsf{EOR}^{F,\varphi}\{u\}, \varphi)$ and therefore $\varphi v\gamma < M$. Using that $N_{v\gamma} \leq \varphi v\gamma < M$ we see that $|C[v]|^{\gamma}_{N_{v\gamma}, W_{v\gamma}}$ is equivalent to

$$W_{v\gamma}(N_{v\gamma}) < u(N_{v\gamma}) \to \bar{B}_0([u](N_{v\gamma}) @ W_{v\gamma}, \mathsf{EOR}^{F,\varphi}([u](N_{v\gamma}) @ W_{v\gamma}))$$

which is true by $P([u](N_{v\gamma}) @ W_{v\gamma})$. By (+) this implies $\bar{B}_0(v, Fv\gamma)$ which by $Fv\gamma \leq \varphi v\gamma < M$ and the fact that $\mathsf{EOR}^{F,\varphi}(u) = Fv\gamma$ implies $\bar{B}_0(u, \mathsf{EOR}^{F,\varphi}(u))$. Therefore by open induction we have $(+) \to \forall u \bar{B}_0(u, \mathsf{EOR}^{F,\varphi}(u))$, from which (†) follows. □

For the full functional interpretation of OI, given by

$$|\mathsf{OI}|^{\Phi,V,\Gamma}_{u,F,W,N} \equiv (|C[V_u]|^{\Gamma_u}_{N_{V_u\Gamma_u}, W_{V_u\Gamma_u}} \to \bar{B}_0(V_u, F_{V_u\Gamma_u})) \to \bar{B}_0(u, \Phi(u))$$

(which is equivalent to $|\mathsf{MBS}^{\mathrm{N}}|^{\Phi,V,\Gamma}_{u,F,W,N}$), we must also extract realizers $V_{u,F,W,N}$ and $\Gamma_{u,F,W,N}$ for v and γ in (†). Let $\Phi_{u,F,W,N} := \mathsf{EOR}^{\varphi^{N,F},F}(u)$ as before, and (suppressing fixed parameters F, W and N on Φ, V, Γ and φ) and let us define candidates for V_u and Γ_u recursively on $<_{\mathrm{lex}}$ by

$$\langle V_u, \Gamma_u \rangle \overset{\sigma^{\mathbb{N}}}{:=} \begin{cases} \Phi\{u\}_{\varphi} & \text{if } |C[\pi_0 \Phi\{u\}_{\varphi}]|^{\pi_1 \Phi\{u\}_{\varphi}}_{N_{\Phi\{u\}_{\varphi}}, W_{\Phi\{u\}_{\varphi}}} \\ \langle V_t, \Gamma_t \rangle & \text{otherwise} \end{cases}$$

for $t := [\pi_0 \Phi\{u\}_{\varphi}](N_{\Phi\{u\}_{\varphi}}) @ W_{\Phi\{u\}_{\varphi}}$. One argues that $\langle V_u, \Gamma_u \rangle$ is well-defined by open induction and Spec. Note that $\Phi\{u\}_{\varphi} = \overline{\Phi\{u\}, M}$ for $M = \mu_{\mathrm{Sp}}(\Phi\{u\}, \varphi)$ so by the fact that $N_{\Phi\{u\}_{\varphi}} \leq \varphi(\Phi\{u\}_{\varphi}) < M$ we have $t := [u](N_{\Phi\{u\}_{\varphi}}) @ W_{\Phi\{u\}_{\varphi}}$, and we only make a recursive call on t in the case that $W_{\Phi\{u\}_{\varphi}}(N_{\Phi\{u\}_{\varphi}}) < u(N_{\Phi\{u\}_{\varphi}})$. Therefore if $\langle V_t, \Gamma_t \rangle$ is well-defined for all $t <_{\mathrm{lex}} u$ then so is $\langle V_u, \Gamma_u \rangle$. The statement that $\langle V_u, \Gamma_u \rangle$ is well-defined is an open property, since $\langle V_u, \Gamma_u \rangle$ depends only on $\overline{u, M}$.

Proposition 8.15. Φ*,* V *and* Γ *as defined above satisfy* $\forall u, F, W, N |\mathsf{OI}|^{\Phi,V,\Gamma}_{u,F,W,N}$, *provably in* $\mathsf{Spec} + \mathsf{OI}$.

Proof. For arbitrary F, W and N we use open induction on the open predicate $P(u) = |\mathsf{OI}|^{\Phi,V,\Gamma}_{u,F,W,N}$. We have only to prove progressiveness. Let us assume that (suppressing the fixed parameters F, W and N as before)

$$(++) \quad |C[V_u]|^{\Gamma_u}_{N_{V_u\Gamma_u},W_{V_u\Gamma_u}} \to \bar{B}_0(V_u, F_{V_u\Gamma_u})$$

holds. We want to derive $\bar{B}_0(u, \Phi(u))$. There are two cases to consider.

Case (a) $|C[\pi_0\Phi\{u\}_\varphi]|^{\pi_1\Phi\{u\}_\varphi}_{N_{\Phi\{u\}_\varphi},W_{\Phi\{u\}_\varphi}}$ holds. Then $\langle V_u, \Gamma_u\rangle = \Phi\{u\}_\varphi$ and $|C[V_u]|^{\Gamma_u}_{N_{V_u\Gamma_u},W_{V_u\Gamma_u}}$ holds, so by (++) we have $\bar{B}_0(\overline{u, M}, F_{\Phi\{u\}_\varphi})$ (for $M = \mu_{\mathrm{Sp}}(\Phi\{u\}, \varphi)$. But $\Phi(u) = \mathsf{EOR}^{\varphi^{N,F},F}(u) = F_{\Phi\{u\}_\varphi}$ and $F_{\Phi\{u\}_\varphi} \leq \varphi(\Phi\{u\}_\varphi) < M$, and therefore we get $\bar{B}_0(u, \Phi(u))$.

Case (b) $|C[\pi_0\Phi\{u\}_\varphi]|^{\pi_1\Phi\{u\}_\varphi}_{N_{\Phi\{u\}_\varphi},W_{\Phi\{u\}_\varphi}}$ fails, which implies:

(i) $\langle V_u, \Gamma_u\rangle = \langle V_t, \Gamma_t\rangle$ for $t = [\pi_0\Phi\{u\}_\varphi](N_{\Phi\{u\}_\varphi}) @ W_{\Phi\{u\}_\varphi} \overset{N_{\Phi\{u\}_\varphi}<M_u}{=} [u](N_{\Phi\{u\}_\varphi}) @ W_{\Phi\{u\}_\varphi}$ with $t(N_{\Phi\{u\}_\varphi}) < u(N_{\Phi\{u\}_\varphi})$;

(ii) $\neg\bar{B}_0(t, \pi_1\Phi\{u\}_\varphi N_{\Phi\{u\}_\varphi} W_{\Phi\{u\}_\varphi})$ and therefore $\neg\bar{B}_0(t, \Phi(t))$ since

$$\begin{aligned}
\pi_1\Phi\{u\}_\varphi N_{\Phi\{u\}_\varphi} W_{\Phi\{u\}_\varphi} &\overset{N_{\Phi\{u\}_\varphi}<M}{=} \pi_1\Phi\{u\} N_{\Phi\{u\}_\varphi} W_{\Phi\{u\}_\varphi} \\
&\overset{W(N)<u(N)}{=} \Phi([u](N_{\Phi\{u\}_\varphi}) @ W_{\Phi\{u\}_\varphi}) \\
&= \Phi(t).
\end{aligned}$$

We can assume $P(t) = |\mathsf{OI}|^{\Phi,V,\Gamma}_{t,F,W,N}$ by hypothesis, and by $\langle V_u, \Gamma_u\rangle = \langle V_t, \Gamma_t\rangle$ and (++) we see that the premise of $|\mathsf{OI}|^{\Phi,V,\Gamma}_{t,F,W,N}$ is true, and therefore we obtain $\bar{B}_0(t, \Phi(t))$. But this contradicts (ii). □

Remark 8.16. We tentatively put forward the idea that this computational interpretation of OI is analogous to the computational interpretation of $\mathsf{TI}_<$ (given via its contrapositive $\mathsf{LEP}_<$) in Lemma 8.6, where in some sense we can identify Kxp defined via transfinite recursion with $\mathsf{EOR}^F(u)$ defined via open recursion, and also $\varepsilon_x(p)$ with $V_u(FNW), \Gamma_u(FNW)$.

We reiterate that this section is only an informal illustration that alternative modes of recursion could potentially be used instead of Spector's bar recursion for obtaining more intuitive programs from classical proofs involving the minimal-bad-sequence argument. However, there are many factors to consider before one can claim in any sense that a mode of recursion is 'better' than Spector's bar recursion for interpreting principles in mathematical analysis, and we leave a more detailed study of the relationship between bar recursion and open recursion to future work.

8.4 Higman's lemma

The minimal bad sequence argument is best known in mathematics for its role in the theory of well-quasi-orders. We call a preorder $(X, \leq_X)$ a *well-quasi-order* (WQO) if any infinite sequence (x_i) in X has the property that $x_i \leq_X x_j$ for some $i < j$. The theory of WQOs contains several results which state that certain constructions on WQOs inherit well-quasi-orderedness, the most famous being Kruskal's tree theorem [59]. A special case of this theorem is what is colloquially known as *Higman's lemma*:

Theorem 8.17 (Higman, [38]). *If $(X, \leq_X)$ is a WQO, then so is the set $(X^*, \leq_{X^*})$ of words in X under the embeddability relation $\leq_{X^*}$, where $\langle x_0, \ldots, x_{m-1}\rangle \leq_{X^*} \langle x'_0, \ldots, x'_{n-1}\rangle$ iff there is a strictly increasing map $f\colon [m] \to [n]$ with $x_i \leq_X x'_{fi}$ for all $i < m$.*

Nash-Williams used the minimal-bad-sequence argument to give elegant proofs of both Higman's lemma and Kruskal's theorem [64], and many proof-theoretic investigations into the minimal-bad-sequence argument tend to focus specifically on its role WQO theory.

We conclude this chapter by demonstrating how one obtains a bound for Higman's lemma from the ND interpretation of the minimal-bad-sequence argument (or equivalently the functional interpretation open induction). By substituting in our realizer from Section 8.2 we obtain an explicit bar recursive term that finds an embedded pair in sequences of words over arbitrary WQOs. We do not claim to derive any new quantitave bounds for Higman's lemma (optimal bounds in certain cases are given in [19, 80]), but we believe that on a qualitative level at least our algorithm, the first based on the functional interpretation, is quite different to those discovered previously.

Formalising Higman's lemma

Let us assume that a WQO $(X, \leq_X)$ can be encoded as by type of level 0, and that $\leq_X$ and therefore also $\leq_{X^*}$ are decidable. We call a sequence $u\colon X^{\mathbb{N}}$ *good* if $u_i \leq_X u_j$ for some $i < j$, and *bad* if it is not good. The statement that u is good can be written as the open predicate $B(u) := \exists k B_0([u](k))$ where

$$B_0(s) :\equiv \exists i < j < |s| (s_i \leq_X s_j).$$

The set X is a WQO if it satisfies the predicate $\mathsf{WQO}[X] :\equiv \forall u^{X^{\mathbb{N}}} B(u)$. In Nash-Williams' proof of Higman's lemma the hypothesis $\mathsf{WQO}[X]$ used in the following, alternative form:

Lemma 8.18. *The property $\mathsf{WQO}[X]$ is classically equivalent to the property $\mathsf{MonSeq}[X]$, where*

$$\mathsf{MonSeq}[X] :\equiv \forall u \exists g^{\mathbb{N}\to\mathbb{N}} \forall k \forall i < j \leq k (gi < gj \wedge u_{gi} \leq_X u_{gj}).$$

In other words, X is a WQO if and only if every infinite sequence contains an infinite increasing subsequence.

Proof. One direction is clear. The other direction $\mathsf{WQO}[X] \to \mathsf{MonSeq}[X]$ is an easy consequence of Ramsey's theorem for pairs, applied to the colouring

$$c(i, j) := 0 \text{ if } u_i \leq_X u_j, \text{ else } 1.$$

Let g be the increasing function that defines a pairwise monochromatic set i.e. $c(gi, gj) = b$ for all $i < j$. If $b = 1$ then this would contradict the fact that X is a WQO, therefore $b = 0$ and g defines a monotone sequence. Note that in special case that X is finite the result is is provable using the infinite pigeonhole principle and can therefore be formalised in $\mathsf{WE\text{-}PA}^\omega + \mathsf{QF\text{-}AC}$. □

Theorem 8.19. $\mathsf{WE\text{-}PA}^\omega + \mathsf{QF\text{-}AC} + \mathsf{MBS} \vdash \mathsf{MonSeq}[X] \to \mathsf{WQO}[X^*]$.

Proof (Nash-Williams). We use the minimal-bad-sequence argument of type $(X^*, \prec)$ where $\prec$ denotes the prefix relation on words:

$$x^{X^*} \prec y :\equiv |x| < |y| \wedge \forall i < |x|(x_i = y_i).$$

Note that $\prec$ is decidable by our assumption that X has type level 0, and is provably well-founded in $\mathsf{WE\text{-}HA}^\omega$, so that in particular $\mathsf{R}_\prec$ is primitive recursively definable. We use the following notation: that for $x = \langle x_0, \ldots, x_{m-1} \rangle \colon X^*$ with $m > 0$

$$\tilde{x} \overset{X^*}{:=} \langle x_0, \ldots, x_{m-2} \rangle \quad \bar{x} \overset{X}{:=} x_{m-1}$$

and for $x = \langle \rangle$ we have $\tilde{x} = \langle \rangle$, $\bar{x} = 0_X$.

Let us assume for contradiction that $\mathsf{MonSeq}[X]$ holds but $\mathsf{WQO}[X^*]$ is not true, and that we have a bad sequence u satisfying $\forall n \neg B_0([u](n))$ where $B_0(s) :\equiv \exists i < j < |s|(s_i \leq_{X^*} s_j)$. Then by $\mathsf{MBS}_{(X^*,\prec)}$ there exists a minimal bad sequence v with respect to $\prec$. We show that v cannot be bad, and therefore our original sequence u was good, contradicting $\neg\mathsf{WQO}[X^*]$.

First, by $\mathsf{MonSeq}[X]$ applied to the sequence $(\bar{v}_i)$ there exists monotone g such that $\bar{v}_{gi} \leq_X \bar{v}_{gj}$ for all $i < j$. Now consider the sequence $w := [v](g0) * (\lambda i.\tilde{v}_{gi})$. Either $v_{g0} = \langle \rangle$ in which case w is trivially good, or $\tilde{v}_{g0} \prec v_{g0}$ and therefore w is lexicographically less than v at point $g0$, and is good by minimality of v.

But if w is good then v must also be good: it is easy to show that if $w_i \leq_{X^*} w_j$ for some $i < j < n$ i.e. $B_0([w](n))$, then $v_{i'} \leq_{X^*} v_{j'}$ for some $i' < j' < g(n)+1$ i.e. $B_0([v](g(n)+1))$. This is obvious if $i < g(0)$. Otherwise we have $\tilde{v}_{gi} \leq_{X^*} \tilde{v}_{gj}$ for $i < j < n - g0 \leq n$, in which case $v_{gi} \leq_{X^*} v_{gj}$ follows from $\bar{v}_{gi} \leq_X \bar{v}_{gj}$, unless $|v_{gi}| = 1$ and $v_{gj} = \langle \rangle$. But in this case $v_{gj} \leq_{X^*} v_{gj+1}$. □

Extracting a realizer for $\mathsf{WQO}[X^*]$

The functional interpretation of $\mathsf{MonSeq}[X]$ (now using the symbol x for the variable u) is given by

$$\exists G \forall x, \omega^{\mathbb{N}^{\mathbb{N}} \to \mathbb{N}} \forall i < j \leq \omega G_{x,\omega}(G_{x,\omega} i < G_{x,\omega} j \wedge x_{G_{x,\omega} i} \leq_X x_{G_{x,\omega} j}) \tag{8.4}$$

for $G \colon X^{\mathbb{N}} \times (\mathbb{N}^{\mathbb{N}} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N})$. The computational challenge we face is that given G witnessing (8.4), and an arbitrary sequence of words u, we must extract $x_{G,u}$, $\omega_{G,u}$ and $\Phi_{G,u}$ such that (8.4) applied to $x_{G,u}$, $\omega_{G,u}$ implies $B_0([u](\Phi_{G,u}))$. We show that this can be done using the functional interpretation of MBS given in the previous section, by constructing suitable counterexample functionals F, N, Y and W.

Theorem 8.20. *Suppose we are given G (dependent on x and ω) satisfying (8.4). Define x_v and $\omega_{v,\gamma}$ parametrised by variables $v \colon (X^*)^{\mathbb{N}}$, $\gamma \colon \mathbb{N} \times (X^*)^{\mathbb{N}} \to \mathbb{N}$ as*

$$\begin{aligned} x_v &\overset{X^{\mathbb{N}}}{:=} (\bar{v}_i), \\ \omega_{v\gamma} &\overset{\mathbb{N}^{\mathbb{N}} \to \mathbb{N}}{:=} \lambda g \,.\, \gamma(g0, [0_{(X^*)^{\mathbb{N}}}](g0) * (\lambda i . \tilde{v}_{gi})). \end{aligned}$$

Now let $G_{v\gamma} := G_{x_v, \omega_{v,\gamma}}$ and define functionals $F \colon \kappa \to \mathbb{N}$, $N \colon \kappa \to \mathbb{N}$ and $W \colon \kappa \to \mathbb{N}$ (for $\kappa = (X^*)^{\mathbb{N}} \times (\mathbb{N} \times (X^*)^{\mathbb{N}} \to \mathbb{N})$ by

$$\begin{aligned} F^G_{v\gamma} &\overset{\mathbb{N}}{:=} G_{v\gamma}(\omega_{v\gamma} G_{v\gamma}) + 1, \\ N^G_{v\gamma} &\overset{\mathbb{N}}{:=} G_{v\gamma} 0, \\ W^G_{v\gamma} &\overset{(X^*)^{\mathbb{N}}}{:=} [0](G_{v\gamma} 0) * (\lambda i . \tilde{v}_{G_{v\gamma} i}). \end{aligned}$$

Then whenever Φ, V and Γ satisfy $|\mathsf{MBS}^{\mathrm{N}}|^{\Phi, V, \Gamma}_{F^G, N^G, W^G, u}$, we have that

$$\forall i < j \leq \omega_{V\Gamma} G_{V\Gamma}(G_{V\Gamma} i < G_{V\Gamma} j \wedge \bar{V}_{G_{V\Gamma} i} \leq_X \bar{V}_{G_{V\Gamma} j})$$

for $V = V_{F^G, N^G, W^G, u}$, $\Gamma = \Gamma_{F^G, N^G, W^G, u}$ implies

$$\exists i < j < \Phi_{F^G, N^G, W^G, u}(u_i \leq_{X^*} u_j).$$

Proof. Let Φ abbreviate $\Phi_{F^G, N^G, W^G, u}$ and V, Γ similarly. Then by $|\mathsf{MBS}^{\mathrm{N}}|^{\Phi, V, \Gamma}_{F^G, N^G, W^G, u}$ (in its contrapositive form) applied to B_0 as in the proof of Theorem 8.19, we have

$$(*) \quad \begin{cases} [(W_{V\Gamma}(N_{V\Gamma}) \prec V(N_{V\Gamma}) \to \bar{B}_0([V](N_{V\Gamma}) \mathbin{@} W_{V\Gamma}, \Gamma N_{V\Gamma} W_{V\Gamma})) \\ \to \bar{B}_0(V, F_{V\Gamma})] \to \bar{B}_0(u, \Phi) \end{cases}$$

and so it suffices to prove the premise of $(*)$, which substituting definitions of F, N and W is

$$\underbrace{(\tilde{V}_{G_{V\Gamma} 0} \prec V_{G_{V\Gamma} 0} \to \bar{B}_0([V](G_{V\Gamma} 0) * (\lambda i . \tilde{V}_{G_{V\Gamma} i}), \omega_{V\Gamma} G_{V\Gamma}))}_{C} \to \bar{B}_0(V, G_{V\Gamma}(\omega_{V\Gamma} G_{V\Gamma}) + 1).$$

Now, we use that $G_{V\Gamma}$ satisfies

$$(\dagger)\ \ \forall i < j \leq \omega_{V\Gamma}G_{V\Gamma}(G_{V\Gamma}i < G_{V\Gamma}j \wedge \bar{V}_{G_{V\Gamma}i} \leq_X \bar{V}_{G_{V\Gamma}j})$$

Assuming C, there are two cases: either $V_{G_{V\Gamma}0} = \langle\rangle$ which implies $\bar{B}_0(V, G_{V\Gamma}(\omega_{V\Gamma}G_{V\Gamma})+1)$ since $G_{V\Gamma}0 \leq G_{V\Gamma}(\omega_{V\Gamma}G_{V\Gamma})$ by (†), or $\tilde{V}_{G_{V\Gamma}0} \prec V_{G_{V\Gamma}0}$.

In the second case we have $\bar{B}_0([V](G_{V\Gamma}0) * (\lambda i.\tilde{V}_{G_{V\Gamma}i}), \omega_{V\Gamma}G_{V\Gamma})$, and using (†) it is straightforward to show, as in the proof of Theorem 8.19, that this implies

$$\bar{B}_0(V, G_{V\Gamma}(\omega_{V\Gamma}G_{V\Gamma}) + 1).$$

This proves the premise of (∗), so we're done. □

Remark 8.21. By combining Theorem 8.20 with Theorem 8.11 we are able to extract an explicit bar recursive realizer in G that bounds the length of bad sequences in X^*. In individual cases the complexity of G will depend on the underlying well-quasi-ordering X. For the case where $X = [m]$ one proves using the infinite pigeonhole principle that any sequence (x_i) of type $X^{\mathbb{N}}$ contains an infinite subsequence $x_{g0} = x_{g1} = \ldots$, and therefore G can be constructed as usual using finite bar recursion (see Lemma 7.14, [67]).

Remark 8.22. Nash-Williams' proof would still work if we carried out the minimal bad sequence argument on the weaker relation $|x| < |y|$, which is decidable even if we allow X to range over higher-type objects (although we still need to insist on $\leq_X$ being decidable).

Remark 8.23. Higman's lemma provides us with a good opportunity to analyse the behaviour of our realizers for MBS in more detail. In this context, the role of our bar recursive realizer from Section 8.2 would be to attempt the construction of a minimal-bad-sequence to obtain a contradiction, from which we can infer that the initial assumption $\neg\bar{B}_0(u, \Phi_{u,F^G,N^G,W^G})$ is false, and therefore Φ_u is a bound for Higman's lemma.

$\mathcal{G}[u, F^G, N^F, W^G]$	Nash-Williams' proof of Higman's lemma
$\tilde{\varepsilon}^u_n$	find least prefix extending to a bad sequence
$\mathsf{EPS}^{\varepsilon}$	attempt to construct a minimal bad sequence
F^G, N^G, W^G	contradict construction of minimal bad sequence
$\Phi(u)$	contradict hypothesis $\neg B(u)$

Table 8.2: Functional interpretation of Higman's lemma

On the other hand, if instead of our bar recursive realizer we use the open recursor $\Phi^G := \lambda u\ .\ \mathsf{EOR}^{F^G,\varphi}(u)$ as in Section 8.3 we end up with an intuitive inductive algorithm that computes the bound $\Phi(u)$ in terms of $\Phi(w)$ for $w <_{\text{lex}} u$, the functionals F^G, N^G and W^G using G to validate well-foundedness of the algorithm.

It would of course be illuminating to give a more detailed description of our realizers, even just for simple cases, and compare them to other programs for Higman's lemma in the literature, but we leave this to future work.

III The Equivalence of Variants of Bar Recursion

CHAPTER NINE

Extensions of Gödel's system T

The second main part of this dissertation is concerned with establishing definability results for variants of bar recursion and related extensions of system T.

In the preceding chapters we encountered Spector's variants of bar recursion GBR and SBR, along with the explicitly iterated product of selection functions EPS. In the remainder of this dissertation the term *bar recursion* refers to a more general class of functionals that can be informally described by the scheme

$$B^T(s) = \begin{cases} G(s) & \text{if } s \text{ is a leaf of } T \\ H(s, \lambda x \, . \, B^T(s * x)) & \text{otherwise} \end{cases}$$

where T is some well-founded tree. Spector's bar recursion is just a specific instance of this type of recursion, over the well-founded tree $T(s) :\equiv \forall t \prec s(|t| \leq \varphi(\widehat{t}))$. A somewhat different instance of bar recursion is the Γ functional of Gandy and Hyland [35]:

$$\Gamma^q(s^{\mathbb{N}^*}) \overset{\mathbb{N}}{:=} q(s * 0 * \lambda n \, . \, \Gamma^q(s * (n+1)))$$

which rather than appealing to an explicit stopping condition as in Spector's bar recursion, is well-founded by assuming continuity of q, and as such its underlying tree is given implicitly via the points of continuity of q. Generalisations of the Γ functional to higher types have been considered in the context of proof theory in order to give a realizability interpretation to the axiom of dependent choice - these include modified bar recursion [11, 12] and more recently the implicitly iterated product of selection functions

$$\mathsf{IPS}^{\varepsilon,q}(s^{\rho^*}) := s \,@\, \lambda n \, . \, \varepsilon_{t_n}(\lambda x \, . \, q(\mathsf{IPS}^{\varepsilon,q}(t_n * x)))$$

for $t_n = [\mathsf{IPS}^{\varepsilon,q}(s)](n)$. A key feature shared by all these variants of bar recursion is that the recursion is carried out 'backwards' over finite sequences, each recursive call extending the current sequence with one more piece of information. An interesting 'demand driven' alternative to bar recursion is the so-called BBC functional:

$$\mathsf{BBC}^{\varepsilon,q}(u) := u \mathbin{@} \lambda n \, . \, \varepsilon_n(\lambda x \, . \, q(\mathsf{BBC}^{\varepsilon,q}(u_n^x))),$$

devised in [8] to give a more efficient computational interpretation to countable choice. The BBC functional takes as input finite partial functions u and makes recursive calls on *updates* u_n^x of u. As demonstrated by Berger [10], the BBC functional belongs to a family of extensions of system T that seem more closely related to open recursion over lexicographic orderings (as discussed in Chapter 8) than bar recursion in the usual sense.

In recent decades a several advances have been made in the computability theory of bar recursion and its variants, particularly in clarifying the relationship between these modes of recursion. In the late 1980s Bezem confirmed that several different formulations of Spector's bar recursion GBR that had been considered in the literature were indeed equivalent [16], while on the other hand Kohenbach devised an interesting a variant of GBR based on a novel stopping condition that could no longer be defined from GBR [46]. Later, Berger and Oliva proved that their modified bar recursion is strictly stronger than Spector's bar recursion due to the fact that the latter is S1-S9 computable in the total continuous functionals $\mathscr{C}^\omega$ while the former S1-S9 defines the FAN functional, which is known to be non-computable in $\mathscr{C}^\omega$. Recently, Escardó and Oliva [28] have given a detailed account of the relationship between bar recursion and the products of selection functions. However, many questions on the interdefinability of variants of bar recursion remain unanswered.

In this dissertation we address some of these open questions. Our contributions are divided into three parts. We begin with a study of finite bar recursion, in which we construct a hierarchy of fragments of system T based on restricted finite bar recursion and prove that they are in one-to-one correspondence with the usual fragments based on restricted primitive recursion (Chapter 10). Next, we show that Spector's 'special' and 'general' form of bar recursion are in fact primitive recursively equivalent (Chapter 11). Finally, we succeed in relating extensions of system T based on open recursion with those based on bar recursion, and prove that over the continuous functionals the BBC functional and open recursion are equivalent to the implicitly iterated product of selection functions (Chapter 12).

In the present chapter we acquaint the reader with some important definitions, and provide a short summary of known facts about extensions of system T in order to put our results in context. Our account is very concise, and the reader is encouraged to consult the original sources where indicated and also e.g. [65] for general background on bar recursion and computability theory in the type structures of continuous functionals.

9.1 Primitive recursive definability

Suppose we state a defining equation $\mathsf{F}(\vec{x}) := \Omega(\mathsf{F}, \vec{x})$ for some mode of recursion F, which contains free variables $\vec{x}$ for the parameters of F. By $\mathsf{F}[\Phi]$ we mean that the term Φ satisfies the defining equation of F i.e.

$$\mathsf{F}[\Phi] :\equiv \forall\vec{x}(\Phi(\vec{x}) = \Omega(\Phi, \vec{x})).$$

We consider F to be an abstract symbol representing a mode of recursion, and technically when we extend some theory S with the functional F, we mean that we add to S a constant Φ along with the axiom $\mathsf{F}[\Phi]$, although usually we implicitly identify Φ with the symbol F.

Definition 9.1. Let S be some theory and Δ a (possibly empty) set of additional axioms in the language of S.

(a) The functional G is S-definable from F ($\mathsf{G} \leq_S \mathsf{F}$) over Δ if there exists a closed term $t \in S$ such that

$$S + \Delta + \mathsf{F}[\Psi] \vdash \mathsf{G}[t(\Psi)].$$

(b) More generally, if we are given defining equations for two collections of functionals $\mathsf{F} \equiv (\mathsf{F}_{\rho_1,\ldots,\rho_k})$, $\mathsf{G} \equiv (\mathsf{G}_{\sigma_1,\ldots,\sigma_l})$ where ρ_i, σ_j range over some subsets X_i, Y_j of the finite types respectively, then G is S-definable from F ($\mathsf{G} \leq_S \mathsf{F}$) over Δ if for any $\sigma_1, \ldots, \sigma_l \colon \Pi_j \mathsf{Y}_\mathsf{j}$ there exists $\rho_1(\sigma_1, \ldots, \sigma_l), \ldots, \rho_k(\sigma_1, \ldots, \sigma_l) \colon \Pi_i \mathsf{X}_\mathsf{i}$ such that

$$\mathsf{G}_{\sigma_1,\ldots,\sigma_l} \leq_S \mathsf{F}_{\rho_1(\sigma_1,\ldots,\sigma_l),\ldots,\rho_k(\sigma_1,\ldots,\sigma_l)}$$

If Δ is empty then we simply say that G is S-definable from F. If $\mathsf{G} \leq_S \mathsf{F}$ and $\mathsf{F} \leq_S \mathsf{G}$ over Δ then F and G are S-equivalent over Δ and we write $\mathsf{F} =_S \mathsf{G}$.

Definition 9.2 (Primitive recursive definability). We say that G is primitive recursively definable from F over Δ if it is $\mathsf{E}\text{-}\mathsf{HA}^\omega$-definable from F over Δ. We will usually write $\leq_\mathsf{T}$ instead of the more cumbersome $\leq_{\mathsf{E}\text{-}\mathsf{HA}^\omega}$.

Remark 9.3. In many cases, when $\mathsf{G} \leq_\mathsf{T} \mathsf{F}$ it turns out that we actually only use a very weak (essentially recursion-free) fragment of $\mathsf{E}\text{-}\mathsf{HA}^\omega$ to construct G from F. However, we don't make any effort to calibrate the weakest theory in which the construction can be done.

The definability relation $\leq_S$ over fixed Δ is clearly transitive. Suppose we are given $\mathsf{F} \equiv (\mathsf{F}_{\rho_1,\ldots,\rho_k})$, $\mathsf{G} \equiv (\mathsf{G}_{\sigma_1,\ldots,\sigma_l})$ and $\mathsf{H} \equiv (\mathsf{H}_{\kappa_1,\ldots,\kappa_m})$ with $\mathsf{G} \leq_S \mathsf{F}$ and $\mathsf{H} \leq_S \mathsf{G}$. Then $\mathsf{H} \leq_S \mathsf{G}$ tells us that

$$S + \Delta + \mathsf{G}_{\sigma_1,\ldots,\sigma_l}[\Phi] \vdash \mathsf{H}_{\kappa_1,\ldots,\kappa_m}[t_0(\Phi)]$$

for some $t_0 \in S$ and $\sigma_i(\kappa_1, \dots, \kappa_m)$, and $\mathsf{G} \leq_S \mathsf{F}$ tells us that

$$S + \Delta + \mathsf{F}_{\rho_1,\dots,\rho_k}[\Psi] \vdash \mathsf{G}_{\sigma_1,\dots,\sigma_l}[t_1(\Psi)]$$

for some $t_1 \in S$ and $\rho_i(\sigma_1, \dots, \sigma_l)$. Putting these together and setting $\Phi := t_1(\Psi)$ we obtain

$$S + \Delta + \mathsf{F}_{\rho_1,\dots,\rho_k}[\Psi] \vdash \mathsf{H}_{\kappa_1,\dots,\kappa_m}[(t_0 \circ t_1)(\Psi))]$$

for $\rho_i(\kappa_1, \dots, \kappa_m) = \rho_i(\sigma_1(\kappa_1, \dots, \kappa_m), \dots, \sigma_l(\kappa_1, \dots, \kappa_m))$. A consequence of this is that the relation $=_S$ is an equivalence relation.

Note that the notion of definability is model independent, although we can talk of definability in a particular model as follows.

Definition 9.4. Let $\mathcal{M}$ be some a model of S. Then G is S-definable from F in $\mathcal{M}$ if $\mathsf{G} \leq_S \mathsf{F}$ over Δ where Δ is valid in $\mathcal{M}$.

In Chapter 12 we shall require $\Delta :\equiv \mathsf{QF\text{-}BI} + \mathsf{Cont}$ to prove that certain implicitly well-founded modes of recursion are primitive recursively equivalent, and so our definability results will in particular be valid in the model $\mathscr{C}^\omega$ of continuous functionals.

9.2 Spector's bar recursion and its variants

In Section 9.3 we introduce, for the first time in this dissertation, some modes of bar recursion which, unlike Spector's bar recursion, do not come with an explicitly defined stopping condition. Before doing so we survey some of the relevant known definability and non-definability results concerning variants of Spector's bar recursion. Let us quickly recall the main definitions from Part II. Spector's general bar recursor $\mathsf{GBR}_{\rho,\tau}$ has defining equation

$$\mathsf{GBR}^{\phi,q,\varphi}_{\rho,\tau}(s^{\rho^*}) \overset{\tau}{:=} \begin{cases} q(s) & \text{if } \varphi(\hat{s}) < |s| \\ \phi_s(\lambda x \ . \ \mathsf{GBR}^{\phi,q,\varphi}(s * x)) & \text{otherwise,} \end{cases}$$

while his special bar recursor SBR_ρ is defined by

$$\mathsf{SBR}^{\phi,\varphi}_{\rho}(s^{\rho^*}) \overset{\rho^{\mathbb{N}}}{:=} s \mathbin{@} \begin{cases} 0_{\rho^{\mathbb{N}}} & \text{if } \varphi(\hat{s}) < |s| \\ \mathsf{SBR}^{\phi,\varphi}_{\rho}(s * a_s) & \text{otherwise} \end{cases}$$

where $a_s := \phi_s(\lambda x \ . \ \mathsf{SBR}(s * x))$. Finally, the explicitly iterated product of selection functions $\mathsf{EPS}_{\rho,\tau}$ is defined as

$$\mathsf{EPS}^{\varepsilon,q,\varphi}(s^{\rho^*}) \overset{\rho^{\mathbb{N}}}{=} s \mathbin{@} \begin{cases} 0_{\rho^{\mathbb{N}}} & \text{if } \varphi(\hat{s}) < |s| \\ \mathsf{EPS}^{\varepsilon,q,\varphi}(s * a_s) & \text{otherwise,} \end{cases}$$

for $a_s := \varepsilon_s(\lambda x \ . \ q(\mathsf{EPS}(s * x)))$. In the above definitions, ρ and τ range over all finite types.

Example 9.5. It is easy to see that EPS is primitive recursively equivalent to SBR, which we prove now as a warm-up example. First we show that $\mathsf{SBR} \leq_{\mathsf{T}} \mathsf{EPS}$. Let

$$\Phi_\rho := t(\mathsf{EPS}_{\rho,\rho^{\mathbb{N}}}) = \lambda\phi, \varphi, s\ .\ \mathsf{EPS}_{\rho,\rho^{\mathbb{N}}}^{\phi,\mathrm{id},\varphi}(s).$$

Note that we technically we mean to define $\Phi = t(\Psi)$ for some variable Ψ satisfying $\mathsf{EPS}_{\rho,\rho^{\mathbb{N}}}[\Psi]$, but for readability we identify Ψ with the symbol EPS. It is easy to see that $\mathsf{E\text{-}HA}^\omega$ proves that $\mathsf{SBR}_\rho[\Phi]$, and therefore $\mathsf{SBR}_\rho \leq_{\mathsf{T}} \mathsf{EPS}_{\rho,\rho^{\mathbb{N}}}$. If $\varphi(\widehat{s}) < |s|$ then $\Phi^{\phi,\varphi}(s) = s \mathbin{@} 0$, and if $\varphi(\widehat{s}) \geq |s|$ then

$$\begin{aligned}\Phi^{\phi,\varphi}(s) &= s \mathbin{@} \mathsf{EPS}^{\phi,\mathrm{id},\varphi}(s * a_s)\\ &= s \mathbin{@} \Phi^{\phi,\varphi}(s * a_s)\end{aligned}$$

where $a_s = \phi_s(\lambda x\ .\ \mathrm{id}(\mathsf{EPS}^{\phi,\mathrm{id},\varphi}(s * x))) = \phi_s(\lambda x\ .\ \Phi^{\phi,\varphi}(s * x))$. Conversely, to prove that $\mathsf{EPS}_{\rho,\tau} \leq_{\mathsf{T}} \mathsf{SBR}_\rho$, let

$$\Phi_{\rho,\tau} := t(\mathsf{SBR}_\rho) = \lambda\varepsilon^{\rho^* \to J_\tau \rho}, q, \varphi, s\ .\ \mathsf{SBR}_\rho^{\phi^{\varepsilon,q},\varphi}(s)$$

where $\phi_s^{\varepsilon,q}(p^{\rho\to\rho^{\mathbb{N}}}) \overset{\rho}{:=} \varepsilon_s(\lambda x\ .\ q(p(x)))$. Then $\varphi(\widehat{s}) < |s|$ implies that $\Phi^{\varepsilon,q,\varphi}(s) = s \mathbin{@} 0$, and $\varphi(\widehat{s}) \geq |s|$ implies

$$\begin{aligned}\Phi^{\varepsilon,q,\varphi}(s) &= s \mathbin{@} \mathsf{SBR}^{\phi^{\varepsilon,q},\varphi}(s * a_s)\\ &= s \mathbin{@} \Phi(s * a_s)\end{aligned}$$

where $a_s = \phi_s(\lambda x\ .\ \mathsf{SBR}(s * x)) = \varepsilon_s(\lambda x\ .\ q(\Phi(s * x)))$. Therefore $\mathsf{EPS}_{\rho,\tau}[\Phi]$ is provable in $\mathsf{E\text{-}HA}^\omega$, from which we conclude $\mathsf{SBR} =_{\mathsf{T}} \mathsf{EPS}$.

Definability between modes of recursion in all finite types may appear to be quite a weak property, in the sense that we impose no restriction on the level of types. To prove that $\mathsf{SBR} \leq_{\mathsf{T}} \mathsf{EPS}$ we could have equally well demonstrated that $\mathsf{EPS}_{\rho^{\mathbb{N}},(\rho^{\mathbb{N}})^{\mathbb{N}}}$ defines SBR_ρ, even though $\mathsf{EPS}_{\rho,\rho^{\mathbb{N}}}$ suffices. However, primitive recursive definability is indeed a non-trivial and subtle property that yields interesting insights into the relationship between extensions of system T. A variant of 'explicitly controlled' bar recursion was given by Kohenbach in [46], which has defining equation

$$\mathsf{KBR}_{\rho,\tau}^{\phi,q,\varphi}(s^{\rho^*}) \overset{\tau}{:=} \begin{cases} q(s) & \text{if } \varphi(\widehat{s}) = \varphi(\check{s})\\ \phi_s(\lambda x\ .\ \mathsf{KBR}_{\rho,\tau}^{\phi,q,\varphi}(s * x)) & \text{otherwise}\end{cases}$$

and is based on a novel stopping condition $\varphi(\widehat{s}) = \varphi(\check{s})$ (where $\check{s} := s \mathbin{@} 1_{\rho^{\mathbb{N}}}$). It turns out that while KBR defines GBR, this adjustment of the stopping condition means that the converse is not true, and $\mathsf{KBR}_{\rho,\tau}$ cannot be defined from $\mathsf{GBR}_{\rho',\tau'}$ for any types ρ', τ'. This is due to the fact that a term Φ satisfying $\mathsf{GBR}[\Phi]$ exists in the majorizable functionals $\mathscr{M}^\omega$, but this is not the case for KBR. For all this see [46]. Note that while KBR will not feature in later chapters here, it plays an important role in the definability results of [11, 12].

Theorem 9.6. *(a)* SBR *is primitive recursively equivalent to* EPS*.*

(b) GBR *primitive recursively defines* SBR*.*

(c) KBR *primitive recursively defines* GBR*, but the converse does not hold.*

Proof. Part (a) is proved in Example 9.5, (b) is similarly straightforward, and for (c) see [46]. □

In Chapter 11 we improve Theorem 9.6 by proving the converse of (b), that GBR is primitive recursively definable from SBR. An important property of these explicitly well-founded modes of bar recursion is that they are computable in the total continuous functionals (see [12] for details).

Theorem 9.7. *Each of* SBR*,* GBR *and* KBR *are S1-S9 computable in the total continuous functionals* $\mathscr{C}^\omega$*.*

Finally, we note that while SBR in general is not primitive recursively definable, the special finite instance of SBR given by setting the control functional φ to be constant:

$$\mathsf{SBR}^{\varepsilon,q}(s) := s \ @ \begin{cases} 0 & \text{if } n < |s| \\ \mathsf{SBR}(s * a_s) & \text{otherwise.} \end{cases}$$

is primitive recursive. Finite bar recursion and its relationship with Gödel's primitive recursor is the main topic of Chapter 10.

9.3 Modified bar recursion and the implicitly controlled product of selection functions

Variants of bar recursion which, rather than relying on a stopping condition as in Spector's bar recursion, are 'implicitly' well-founded by a continuity argument have been used in proof theory to give a modified realizability interpretation to countable dependent choice. Modified bar recursion, developed in [11, 12] (and in turn based on a form of bar recursion considered in [8]), is given by the defining equation

$$\mathsf{MBR}^{\psi,q}_{\rho,\tau}(s^{\rho^*}) \overset{\tau}{:=} q(s \ @ \ \psi_s(\lambda x^\rho \,.\, \mathsf{MBR}(s * x)))$$

for $\psi\colon \rho^* \to ((\rho \to \tau) \to \rho^{\mathbb{N}})$ and $q\colon \rho^{\mathbb{N}} \to \tau$. Provided τ is restricted to being a discrete type, MBR exists in the model of continuous functionals $\mathscr{C}^\omega$ by the continuity principle Cont (see [11]). In what follows we focus on an alternative formulation of implicitly controlled bar recursion considered by Escardó and Oliva called the implicitly controlled product of selection functions, given by the defining equation

$$\mathsf{IPS}^{\varepsilon,q}_{\rho,\tau}(s^{\rho^*}) \overset{\rho^{\mathbb{N}}}{:=} s \ @ \ \mathsf{IPS}^{\varepsilon,q}_{\rho,\tau}(s * a_s) \tag{9.1}$$

for $a_s := \varepsilon_s(\lambda x\ .\ q(\mathsf{IPS}^{\varepsilon,q}_{\rho,\tau}(s * x)))$, where $\varepsilon\colon \rho^* \to J_\tau\rho$, $q\colon \rho^{\mathbb{N}} \to \tau$ and τ is restricted to being discrete. It is shown in [28] that IPS is equivalent to MBR, although IPS has the advantage of providing us with a uniform transition between explicitly and implicitly defined bar recursion in that it is essentially EPS without the stopping condition. As such it also satisfies a useful analogue of Theorem 5.7.

Lemma 9.8. *Let $\alpha = \mathsf{IPS}^{\varepsilon,q}(s)$. Then for all $n \geq |s|$,*

$$\alpha = \mathsf{IPS}^{\varepsilon,q}([\alpha](n)). \tag{9.2}$$

Proof. Induction on n. For $n = |s|$ we have $[\alpha](n) = [\mathsf{IPS}(s)](|s|) = s$ by definition, and therefore (9.2) holds by definition. Now assume that (9.2) holds for $n \geq |s|$. Then

$$\alpha \overset{\text{IH}}{=} \mathsf{IPS}([\alpha](n)) = [\alpha](n) \mathbin{@} \mathsf{IPS}([\alpha](n) * a) \overset{(*)}{=} \mathsf{IPS}([\alpha](n) * a)$$

where $a = \varepsilon_{[\alpha](n)}(\lambda x\ .\ q(\mathsf{IPS}([\alpha](n) * x)))$ and $(*)$ holds because $[\alpha](n)$ is a prefix of $\mathsf{IPS}([\alpha](n) * a)$. Now $\alpha(n) = \mathsf{IPS}([\alpha](n) * a)(n) = a$ and therefore $\alpha = \mathsf{IPS}([\alpha](n) * \alpha(n)) = \mathsf{IPS}([\alpha](n+1))$. □

Theorem 9.9. *Define $\alpha\colon \rho^{\mathbb{N}}$ and $p_s\colon \rho \to \tau$ by*

$$\begin{aligned} \alpha &:= \mathsf{IPS}^{\varepsilon,q}_{\rho,\tau}(\langle\rangle) \\ p_s &:= \lambda x\ .\ q(\mathsf{IPS}^{\varepsilon,q}_{\rho,\tau}(s * x)). \end{aligned}.$$

Then for all n we have

$$\begin{aligned} \alpha(n) &= \varepsilon_{[\alpha](n)}(p_{[\alpha](n)}) \\ p_{[\alpha](n)}(\alpha(n)) &= q(\alpha). \end{aligned} \tag{9.3}$$

Proof. For the first inequality we have

$$\begin{aligned} \alpha(n) &\overset{L9.8}{=} \mathsf{IPS}([\alpha](n))(n) \\ &\overset{(9.1)}{=} ([\alpha](n) \mathbin{@} \mathsf{IPS}([\alpha](n) * a_{[\alpha](n)}))(n) \\ &\overset{(9.1)}{=} a_{[\alpha](n)} \end{aligned}$$

where $a_{[\alpha](n)} = \varepsilon_{[\alpha](n)}(\lambda x\ .\ q(\mathsf{IPS}([\alpha](n) * x))) = \varepsilon_{[\alpha](n)}(p_{[\alpha](n)})$. For the second, we have

$$\begin{aligned} q\alpha &\overset{L9.8}{=} q(\mathsf{IPS}([\alpha](n+1))) \\ &= q(\mathsf{IPS}([\alpha](n) * \alpha(n))) \\ &= p_{[\alpha](n)}(\alpha(n)). \end{aligned}$$

□

The equations (9.3) are of course analogous to Spector's equations for EPS. A class of sequential games for which IPS computes optimal strategies is discussed in [31]. In the remaining chapters we are no longer concerned with the semantics of the products of selection functions, but Lemma 9.8 and Theorem 9.9 will be used in Chapter 12 to prove that IPS defines open recursion.

If we unwind the definition of IPS we can view it in an alternative form in which it is given using course of values recursion.

Proposition 9.10. *The product of selection functions* IPS *can be equivalently defined by the equation*

$$\mathsf{IPS}^{\varepsilon,q}(s) := s \mathbin{@} \lambda n \, . \, \varepsilon_{t_n}(\lambda x \, . \, q(\mathsf{IPS}^{\varepsilon,q}(t_n * x))) \tag{9.4}$$

where $t_n := [\mathsf{IPS}(s)](n)$*.*

Proof. Technically what we mean is that any functional Φ satisfying (9.1) also satisfies (9.4), and vice-versa, provably in $\mathsf{E\text{-}HA}^\omega$. The proof of this fact is straightforward and can be found in Appendix A. □

While it is easy to verify that IPS defines a total continuous functional by the continuity axiom applied to q (recalling that the outcome type τ of q is restricted to being discrete), it turns out that it is *not* S1-S9 computable in the total continuous functionals $\mathscr{C}^\omega$. This follows from the equivalent fact that MBR is not S1-S9 definable in $\mathscr{C}^\omega$, which was proved in [11] using properties of the FAN functional. The FAN functional is a functional of type $(\mathbb{B}^\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ that computes a point of uniform continuity for continuous functionals on on the Cantor space, i.e. has defining axiom

$$\mathsf{FAN}[\Phi] \; : \; \forall q^{\mathbb{B}^\mathbb{N} \to \mathbb{N}}, \alpha, \beta([\alpha](\Phi(q)) = [\beta](\Phi(q)) \to q(\alpha) = q(\beta)).$$

While FAN exists in $\mathscr{C}^\omega$, it is well-known that the FAN functional is not S1-S9 computable in the total continuous functionals (see [35, 65]).

Theorem 9.11 (Berger/Oliva [11, 12])**.** *The* FAN *functional is primitive recursively definable from* $\mathsf{KBR}+\mathsf{MBR}$ *in* $\mathsf{Cont}+\mathsf{QF\text{-}BI}$*, and therefore by Theorem 9.7 is S1-S9 definable from* MBR *in* $\mathscr{C}^\omega$*. But this means that* MBR *is not S1-S9 computable in* $\mathscr{C}^\omega$*.*

We now summarise the known definability results concerning MBR and IPS in the following theorem.

Theorem 9.12. *(a)* IPS *and* MBR *are primitive recursively equivalent over* $\mathsf{QF\text{-}BI}+\mathsf{Cont}$*.*

(b) Both IPS *and* MBR *primitive recursively define* GBR *over* $\mathsf{QF\text{-}BI}+\mathsf{Cont}$*.*

(c) MBR *does not define* KBR*.*

(d) *None of* SBR, GBR *or* KBR *primitive recursively define* IPS *or* MBR *over* QF-BI+Cont *or any theory validated by* $\mathscr{C}^\omega$.

Proof. (a) This is proved in [28] using a slightly different formulation sIPS of IPS. We confirm in Appendix A that sIPS and IPS are equivalent over QF-BI + Cont.

(b) Definability of GBR from MBR is given in [12]. Definability of GBR from IPS follows from part (a) or is proven directly in [28].

(c) Proved in [12] by the fact that MBR exists in $\mathscr{M}^\omega$.

(d) By Theorem 9.11 and the fact that S1-S9 computable functionals in $\mathscr{C}^\omega$ are closed under primitive recursion, MBR is not definable from any S1-S9 computable functional over any theory Δ validated by $\mathscr{C}^\omega$, so the result follows by Theorem 9.7. See [12] for details. An analogous result follows for IPS (and indeed any functional that defines MBR over $\mathscr{C}^\omega$).

□

9.4 The Berardi-Bezem-Coquand functional and update recursion

In [8], Berardi, Bezem and Coquand consider two different computational interpretations of choice, for AC and DC respectively. The latter is the basis of modified bar recursion as formalised in [11]. The former, on the other hand, is a remarkable form of recursion that constructs a sequence using a demand-driven, symmetric algorithm that is quite different to bar recursion. We call this the BBC functional.

In order to define BBC we require a few preliminary definitions. We view the type $\bar{\rho}^{\mathbb{N}}$ where $\bar{\rho} :\equiv \mathbb{B} \times \rho$ as the type of *partial sequences* over ρ. A partial sequence $u\colon \bar{\rho}^{\mathbb{N}}$ is defined at n, or n is in the domain of u, if and only if $u(n)_0 = 1$. We define the decidable predicates $n \in \mathrm{dom}(u) :\equiv (u(n)_0{=}1)$ and $n \notin \mathrm{dom}(u) :\equiv \neg(n \in \mathrm{dom}(u)) \leftrightarrow (u(n)_0 = 0)$. If $n \in \mathrm{dom}(u)$ then the value of u at n is $u(n)_1\colon \rho$. We define $u[n] :\equiv u(n)_1$. We redefine the overwrite operator @ in the context of partial sequences as the functional @ : $\bar{\rho}^{\mathbb{N}} \times \rho^{\mathbb{N}} \to \rho^{\mathbb{N}}$ defined by

$$(u \mathbin{@} \alpha)(n) \overset{\rho}{:=} \begin{cases} u[n] & \text{if } n \in \mathrm{dom}(u) \\ \alpha(n) & \text{otherwise.} \end{cases}$$

It will always be clear from the context which @ we are using. Finally, given a partial sequence $u\colon \bar{\rho}^{\mathbb{N}}$ and $x\colon \rho$, the partial sequence u_n^x is defined by

$$u_n^x(m) \overset{\bar{\rho}}{:=} \begin{cases} (1, x) & \text{if } m = n \\ u(m) & \text{otherwise.} \end{cases}$$

We say that u_n^x is an *update* of u whenever $n \notin \mathrm{dom}(u)$. The functional $\mathsf{BBC}_{\rho,\tau}$ is given by the defining equation

$$\mathsf{BBC}_{\rho,\tau}^{\varepsilon,q}(u^{\bar{\rho}^{\mathbb{N}}}) \overset{\rho^{\mathbb{N}}}{:=} u \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x^\rho \,.\, q(\mathsf{BBC}^{\varepsilon,q}(u_n^x))),$$

where $\varepsilon\colon \mathbb{N} \to J_\tau\rho$, $q\colon \rho^{\mathbb{N}} \to \tau$ and τ is restricted to being discrete. Originally, BBC was defined in a slightly different form sBBC.

Proposition 9.13. *The functional* $\mathsf{BBC}_{\rho,\tau}$ *is primitive recursively equivalent to* $\mathsf{sBBC}_{\rho,\tau}$ *defined by*

$$\mathsf{sBBC}_{\rho,\tau}^{\varepsilon,q}(u) \overset{\tau}{:=} q(u \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, \mathsf{sBBC}^{\varepsilon,q}(u_n^x))).$$

Proof. Define $\Phi^{\varepsilon,q}(u) := q(\mathsf{BBC}^{\varepsilon,q}(u))$. Then $\mathsf{sBBC}[\Phi]$ follows because

$$\begin{aligned}\Phi^{\varepsilon,q}(u) &= q(u \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, q(\mathsf{BBC}^{\varepsilon,q}(u_n^x))))\\ &= q(u \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, \Phi^{\varepsilon,q}(u_n^x))).\end{aligned}$$

Conversely, by defining $\Phi^{\varepsilon,q}(u) := u \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, \mathsf{sBBC}^{\varepsilon,q}(u_n^x))$ one obtains a functional satisfying $\mathsf{BBC}[\Phi]$:

$$\begin{aligned}\Phi^{\varepsilon,q}(u) &= u \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, q(u_n^x \mathbin{@} \lambda m \,.\, \varepsilon_m(\lambda y \,.\, \mathsf{sBBC}^{\varepsilon,q}((u_n^x)_m^y))))\\ &= u \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, q(\Phi^{\varepsilon,q}(u_n^x))).\end{aligned}$$

□

Remark 9.14. Whenever u has finite domain, the computation of $\mathsf{BBC}(u)$ only ever makes recursive calls on partial sequences with finite domain - in particular $\mathsf{BBC}(\langle\rangle)$ is defined in terms of BBC restricted to arguments with finite domain. In fact BBC was originally defined purely in terms of partial sequences with finite domain (see [8, 9]) which can be encoded as finite sequences of type $(\mathbb{N}\times\rho)^*$. Thus our definition of BBC is slightly different to some definitions found in the literature (notably the original paper [8] itself), although both formulations are discussed in [9, 10].

Berger demonstrated that BBC is easier to understand and its totality elegantly proven when it is viewed as a special instance of *update recursion*. Update recursion is the computational content of the principle of update induction UI, which is defined in [10] as the schema

$$\mathsf{UI} \;:\; \forall w^{\bar{\rho}^{\mathbb{N}}}(\forall w{<_{\mathrm{up}}}vB(w) \to B(v)) \to \forall uB(u)$$

where B ranges over open predicates (cf. Chapter 8) on partial sequences of type $\bar{\rho}^{\mathbb{N}}$ and $w{<_{\mathrm{up}}}v$ iff w is an update of v. Accordingly, update recursion has defining equations

$$\mathsf{UR}_{\rho,\tau}^{G}(u^{\bar{\rho}^{\mathbb{N}}}) \overset{\tau}{:=} G_u(\lambda n, x^\rho \,.\, \mathsf{UR}^G(u_n^x) \text{ if } n \notin \mathrm{dom}(u))$$

where $G\colon \bar{\rho}^{\mathbb{N}} \times (\mathbb{N} \times \rho \to \tau) \to \tau$ and τ is restricted to being discrete.

Theorem 9.15 (Berger [10])**.** *Update recursion exists in the total continuous functionals $\mathscr{C}^\omega$.*

Proof. Update recursion can be defined in the partial continuous functionals $\hat{\mathscr{C}}^\omega$ as a suitable fixpoint. To prove totality of UR in $\hat{\mathscr{C}}^\omega$ (and therefore the existence of UR in $\mathscr{C}^\omega$), the basic idea is that given total G, the predicate 'u total $\to \mathsf{UR}^G(u)$ total' is equivalent to an open predicate $B(u)$ by the continuity axiom. Now clearly $B(u)$ is update progressive by definition of UR, therefore by update induction we have that UR^G must be total. □

In [10] it is shown that UI implies $\mathsf{AC}^{\mathbb{N}}$, and moreover that the update-recursive program extracted from this proof is essentially the BBC functional, so in particular UR defines the BBC functional.

Proposition 9.16. *The* BBC *functional is primitive recursively definable from* UR.

Proof. Define $\Phi^{\varepsilon,q}(u) := \mathsf{UR}^{G^{\varepsilon,q}}_{\rho,\tau}(u)$ where

$$G^{\varepsilon,q}_u(P^{\mathbb{N}\times\rho\to\tau}) \overset{\tau}{:=} q(u \mathbin{@} \lambda n\,.\,\varepsilon_n(\lambda x\,.\,Pnx)).$$

Then we have $\mathsf{sBBC}_{\rho,\tau}[\Phi]$ because

$$\begin{aligned}\Phi^{\varepsilon,q}(u) &= G^{\varepsilon,q}_u(\lambda n, x\,.\,\mathsf{UR}^{G^{\varepsilon,q}}(u^x_n) \text{ if } n \notin \mathrm{dom}(u))\\ &\overset{(*)}{=} q(u \mathbin{@} \lambda n\,.\,\varepsilon_n(\lambda x\,.\,\mathsf{UR}^{G^{\varepsilon,q}}(u^x_n)))\\ &= q(u \mathbin{@} \lambda n\,.\,\varepsilon_n(\lambda x\,.\,\Phi^{\varepsilon,q}(u^x_n))),\end{aligned}$$

where $(*)$ follows from the fact that $u \mathbin{@} \lambda n\,.\,\varepsilon_n(\lambda x\,.\,\mathsf{UR}^{G^{\varepsilon,q}}(u^x_n))$ only makes recursive calls on $\mathsf{UR}^{G^{\varepsilon,q}}(u^x_n)$ for $n \notin \mathrm{dom}(u)$. □

Corollary 9.17. *The* BBC *functional exists in the total continuous functionals* $\mathscr{C}^\omega$.

Remark 9.18. The defining equations for BBC are similar to those of IPS, but the two functionals differ in a fundamental way. The product of selection functions makes recursive calls in sequential manner - the value of $\mathsf{IPS}(s)(n)$ depends on $\mathsf{IPS}(s)(m)$ for all $m < n$. On the other hand, BBC makes recursive calls symmetrically, so that for $m \neq n$, $\mathsf{BBC}(u)(n)$ and $\mathsf{BBC}(u)(m)$ are computed independently over distinct computation trees.

However, the order of computation is essentially the only way in which they differ. Indeed, both arise from the same construction - which we illustrate very informally as follows. Suppose that $\prec$ organises $\mathbb{N}$ into a well-founded tree i.e. the set $\mathrm{pd}(n)$ of predecessors of n is finite and well-ordered, and define $\mathsf{F}_\prec$ on $\prec$-closed partial functions $u\colon \mathbb{N} \to \rho$ by

$$\mathsf{F}^{\varepsilon,q}_\prec(u) \overset{\rho^{\mathbb{N}}}{:=} u \mathbin{@} \lambda n\,.\,\varepsilon_{n,S_n}(\lambda x\,.\,q(\mathsf{F}_\prec(u \mathbin{@} (S_n \cup \langle n, x\rangle))))$$

where $S_n := \{\langle m, \mathsf{F}_\prec(u)(m)\rangle \mid m \in \mathrm{pd}(n)\}$ and the partial function $S_n \cup \langle n, x\rangle$ is defined in the obvious sense.

Then when $\prec$ is the usual order $<$, we can identify $<$-closed partial functions with finite sequences ρ^*, S_n with $t_n = [\mathsf{F}_<(s)](n)$ and $s @ (t_n \cup \{n, x\}) = t_n * x$ for $n \geq |s|$, so the defining equations of F are essentially those for IPS.

Similarly, when $\prec$ is the discrete order $\bullet$, $<$-closed partial functions are just partial functions in the usual sense, S_n is empty, and $u @ (\emptyset \cup \{n, x\}) = u_n^x$ for $n \notin \mathrm{dom}(u)$, so the defining equation for $\bullet$ is the defining equation for BBC.

9.5 Open recursion

The final form of recursion that we study in this part of the dissertation is *open recursion*, which has already been discussed in Chapter 8 in terms of the computational content of open induction. It is given by the defining equation

$$\mathsf{OR}^F_{(\rho,<),\tau}(u^{\rho^{\mathbb{N}}}) \overset{\tau}{:=} F_u(\lambda n, y, v \,.\, \mathsf{OR}^F([u](n) * y @ v) \text{ if } y < u(n))$$

where $F\colon \rho^{\mathbb{N}} \times (\mathbb{N} \times \rho \times \rho^{\mathbb{N}} \to \tau) \to \tau$, τ is discrete and $<$ is a decidable well-founded relation on ρ (note that unlike Chapter 8 we make the variable y explicit, so that now OI is exactly as defined in [10]).

Theorem 9.19 (Berger [10])**.** *Open recursion exists in the total continuous functionals* $\mathscr{C}^\omega$*.*

Proof. See [10, Proposition 5.1]. This follows in analogous fashion to the proof that update induction is total, although this time we appeal to open induction. □

It is easily seen (as pointed out in [10]) that update induction is an instance of open induction on sequences of type $\bar{\rho}^{\mathbb{N}}$ with the well-founded relation $(a, x) < (b, y) :\equiv (a = 1 \wedge b = 0)$. It follows immediately from this observation that update recursion is an instance of open recursion.

Proposition 9.20. *Update recursion is primitive recursively definable from open recursion.*

Proof. Given $G\colon \bar{\rho}^{\mathbb{N}} \times (\mathbb{N} \times \rho \to \tau) \to \tau$ define

$$F^G_u(P^{\mathbb{N}\times\bar{\rho}\times\bar{\rho}^{\mathbb{N}}\to\tau}) \overset{\tau}{:=} G_u(\lambda n, x \,.\, Pn(1, x)u).$$

Then $\Phi^G(u) := \mathsf{OR}^{F^G}_{\bar{\rho},\tau,<}(u)$ where $<$ is defined as above satisfies $\mathsf{UR}_{\rho,\tau}[\Phi]$ because

$$\begin{aligned}
\Phi^G(u) &= F^G_u(\lambda n, y, v \,.\, \mathsf{OR}^{F^G}([u](n) * y @ v) \text{ if } y < u(n)) \\
&= G_u(\lambda n, x \,.\, \mathsf{OR}^{F^G}([u](n) * (1, x) @ u) \text{ if } (1, x) < u(n)) \\
&\overset{(*)}{=} G_u(\lambda n, x \,.\, \mathsf{OR}^{F^G}(u_n^x) \text{ if } n \notin \mathrm{dom}(u)) \\
&= G_u(\lambda n, x \,.\, \Phi^G(u_n^x) \text{ if } n \notin \mathrm{dom}(u)).
\end{aligned}$$

Note that we have $(*)$ because $(1, x) < u(n) \leftrightarrow u(n)_0 = 0 \leftrightarrow n \notin \mathrm{dom}(u)$ and $[u](n) * (1, x) @ u = u_n^x$ (note that here $@$ denotes overwrite on *sequences*). □

9.6 Summary of interdefinability results

Figure 9.1 summarises the known definability results between the extensions of system T discussed in this section, and highlights the new results that will be presented in the following chapters with dotted lines. Here $\Delta :\equiv \mathsf{QF\text{-}BI} + \mathsf{Cont}$. We prove that:

- Finite bar recursion fP_n of type level n is equivalent to Gödel's primitive recursion R_{n+1} of type level $n+1$ (Chapter 10);
- Spector's special bar recursion SBR defines his general bar recursion GBR, and so the two are in fact equivalent (Chapter 11);
- The implicitly iterated product of selection functions defines open recursion, and conversely the BBC functional defines IPS, both over Δ (Chapter 12). This means that IPS, MBR, OR, UR and BBC all lie in the same equivalence class over Δ.

$$\begin{tikzcd}
 & & \mathsf{OR} \arrow[r] & \mathsf{UR} \arrow[d] \\
 & \mathsf{MBR} \arrow[r, leftrightarrow, "\Delta"] \arrow[dl, bend right, "\shortmid" marking] \arrow[d, bend left, "\Delta"] & \mathsf{IPS} \arrow[u, dotted, "\Delta", "\text{Ch. 12}"'] & \mathsf{BBC} \arrow[l, dotted, "\text{Ch. 12}"', "\Delta"] \\
\mathsf{KBR} \arrow[ur, bend left, "\Delta", "\shortmid" marking] \arrow[r, bend left] & \mathsf{GBR} \arrow[l, bend left, "\shortmid" marking] \arrow[u, bend left, "\Delta", "\shortmid" marking] \arrow[r, bend left] & \mathsf{SBR} \arrow[l, dotted, bend left, "\text{Ch. 11}"] \arrow[r, leftrightarrow] & \mathsf{EPS} \arrow[d, bend right] \\
 & & \mathsf{R}_{n+1} & \mathsf{fP}_n \arrow[u, bend right, "\Delta"', "\shortmid" marking] \arrow[l, dotted, leftrightarrow, "\text{Ch. 10}"]
\end{tikzcd}$$

Figure 9.1: Summary of definability results

CHAPTER TEN

Finite Bar Recursion and the Primitive Recursive Functionals

In [66] Oliva demonstrated that an elegant description of the primitive recursive realizer for the functional interpretation of the infinite pigeonhole principle IPHP is given by a finite form of Spector's bar recursion. The reason for this is that the infinite pigeonhole principle is a direct consequence of the axiom of bounded collection, whose negative translation follows from the finite double negation shift - a principle naturally interpreted by finite bar recursion.

In this chapter we study finite bar recursion and its relationship to Gödel's primitive recursion. It is clear that unlike the usual bar recursion, finite bar recursion is primitive recursively definable. We prove the converse: that finite bar recursion defines, essentially over the λ-calculus, all primitive recursive functionals. More specifically we define a hierarchy P_n of fragments of system T based on finite bar recursors of restricted type, and show that these fragments are in one-to-one correspondence with the usual fragments T_{n+1} based on primitive recursion.

We then clarify how finite bar recursion can be used to interpret fragments of arithmetic based on bounded collection or finite choice principles. Parsons demonstrated that the induction based fragment of arithmetic $\mathsf{I}\Sigma_n$ has a functional interpretation in the fragment T_{n-1} of system T [73]. We provide an analogous result for choice-based fragments of arithmetic and bar recursive fragments of T.

We believe that our alternative construction of system T has several key benefits.

1. Finite bar recursion is equivalent to a finite product of selection functions which can be viewed as a functional that computes optimal strategies in finite sequential games. This correspondence gives us an illuminating characterisation of the primitive recursive functionals. In particular the fragment P_n, equivalent to T_{n+1}, is built from functionals that compute optimal strategies in sequential games whose moves have type level n.

2. Finite bar recursion can be used as an alternative to (or in conjunction with) primitive recursion in order to extract readable and intuitive programs from proofs in arithmetic that involve bounded collection or finite choice principles.

3. The equivalence of the finite product with the recursor allows for a smooth transition from the functional interpretation of arithmetic (product with a fixed number of iterations) to that of analysis (product with a finite but unbounded number of iterations). Therefore, we obtain the correspondence

$$\frac{\text{Finite product of selection functions}}{\text{Arithmetic}} = \frac{\text{Unbounded product of selection functions}}{\text{Analysis}}$$

10.1 Finite bar recursion and its fragments

We begin by constructing a new hierarchy of fragments of T based on finite bar recursion. First, let's recall the standard hierarchy based on primitive recursion.

Definition 10.1. System T consists of the terms of $\mathsf{E}\text{-}\mathsf{HA}^\omega$. The fragment T_n consists the subclass of T with recursion R_ρ restricted to types with $\deg(\rho) \leq n$.

Definability results between primitive recursive functionals will be officially carried out in the lowest fragment T_0 of T as we require basic recursive operations on sequences and definition by cases, although it is clear that a weak fragment of T_0 suffices for all the constructions that follow. Similarly, we allow ourselves the whole of $\mathsf{E}\text{-}\mathsf{HA}^\omega$ to verify our results and make no effort to establish the weakest fragment of Heyting arithmetic over which they can be proved.

The finite bar recursor fB_ρ has defining equation

$$\mathsf{fB}^{\phi,n}_\rho(s^{\rho^*}) \overset{\rho^*}{:=} s \mathbin{@} \begin{cases} \langle\rangle & \text{if } |s| > n \\ \mathsf{fB}^{\phi,n}(s * a_s) & \text{if } |s| \leq n \end{cases}$$

for $a_s := \phi_s(\lambda x \mathbin{.} \mathsf{fB}^{\phi,n}(s * x))$, where $\phi \colon \rho^* \to J_{\rho^*}\rho$.

Definition 10.2. The fragment P_n of T consists of T_0 extended with the finite bar recursor fB_ρ restricted to types with $\deg(\rho) \leq n$.

Similarly to fB_ρ, the finite product of selection functions $\mathsf{fP}_{\rho,\tau}$ has defining equation

$$\mathsf{fP}^{\varepsilon,q,n}_{\rho,\tau}(s) \overset{\rho^*}{:=} s \mathbin{@} \begin{cases} \langle\rangle & \text{if } |s| > n \\ \mathsf{fP}^{\varepsilon,q,n}(s * a_s) & \text{if } |s| \leq n \end{cases}$$

for $a_s := \varepsilon_s(\lambda x \mathbin{.} q(\mathsf{fP}(s * x)))$, where $\varepsilon \colon \rho^* \to J_\tau\rho$ and $q \colon \rho^* \to \tau$.

Proposition 10.3. *The finite bar recursor* fB_ρ *defines* $\mathsf{fP}_{\rho,\tau}$ *for arbitrary* τ*, and consequently the fragment* P_n *contains* $\mathsf{fP}_{\rho,\tau}$ *for* τ *arbitrary and* $\deg(\rho) \leq n$*. Conversely the finite product* $\mathsf{fP}_{\rho,\rho^*}$ *defines* fB_ρ*, therefore one can alternatively define* P_n *as* T_0 *plus* $\mathsf{fP}_{\rho,\tau}$ *for* $\deg(\rho) \leq n$*.*

Proof. This follows entirely analogously from the equivalence of the corresponding unbounded bar recursors (see Example 9.5). □

By Proposition 10.3 we will use fB_ρ and $\mathsf{fP}_{\rho,\tau}$ interchangeably in the remainder of this Chapter. From a straightforward adaptation of Theorem 5.7 one easily derives a finitary analogue for the main theorem on EPS for fP and fB.

Proposition 10.4. *If* $s = \mathsf{fP}^{\varepsilon,q,n}(\langle\rangle)$ *and* $p_t = \lambda x \, . \, q(\mathsf{fP}^{\varepsilon,q,n}(t * x))$ *then for all* $i \leq n$ *we have*

$$\begin{aligned} s_i &= \varepsilon_{[s](i)}(p_{[s](i)}) \\ p_{[s](i)}(s_i) &= q(s). \end{aligned}$$

Because $\mathsf{fB}^{\phi,n}_\rho$ *is just* $\mathsf{fP}^{\phi,id,n}_{\rho,\rho^*}$*, note that if* $s = \mathsf{fB}^{\phi,n}(\langle\rangle)$ *and* $p_t = \lambda x \, . \, \mathsf{fB}^{\phi,n}(t * x)$ *then similarly*

$$\begin{aligned} s_i &= \phi_{[s](i)}(p_{[s](i)}) \\ p_{[s](i)}(s_i) &= s. \end{aligned}$$

for $i \leq n$*.*

Finite bar recursion was originally defined slightly differently in [66], as a functional sfB with defining equation

$$\mathsf{sfB}^{\varepsilon,\Delta,n}_{\rho,\tau}(s) \overset{\rho^*}{:=} \begin{cases} \langle\rangle & \text{if } |s| > n \\ X_s * \mathsf{sfB}(s * X_s) & \text{if } |s| \leq n \end{cases}$$

for $X_s := \varepsilon_{|s|}(\lambda x \, . \, \Delta_{|s|}(s * x * \mathsf{sfB}(s * x)))$, where $\varepsilon \colon \mathbb{N} \to J_\tau \rho$ and $\Delta \colon \mathbb{N} \to (\rho^* \to \tau)$. While quite similar to $\mathsf{fP}_{\rho,\tau}$, it differs in the (obviously inessential) fact that it only returns the tail of the recursion, and more significantly in that the selection functions are not allowed access to previously computed values s. Since finite bar recursion has often been given in this form in the literature, we include the following result.

Proposition 10.5. fB_ρ *defines* $\mathsf{sfB}_{\rho,\tau}$ *for arbitrary* τ*, and conversely* $\mathsf{sfB}_{\rho,\rho^*}$ *defines* fB_ρ*.*

Proof. Firstly, define $\Phi^{\varepsilon,\Delta,n}(s) := \mathsf{tail}_{|s|}(\mathsf{fB}^{\phi^{\varepsilon,\Delta},n}_\rho(s))$ where $\mathsf{tail}_n(t)$ denotes the finite sequence t without its first n elements (or just $\langle\rangle$ whenever $|t| \leq n$) and

$$\phi^{\varepsilon,\Delta}_s(p^{\rho \to \rho^*}) := \varepsilon_{|s|}(\lambda x \, . \, \Delta_{|s|}(Px)).$$

Then we have $\mathsf{sfB}_{\rho,\tau}[\Phi]$. If $|s| > n$ then $\Phi^{\varepsilon,\Delta,n}(s) = \mathsf{tail}_{|s|}(s) = \langle\rangle$, and otherwise (suppressing parameters)

$$\begin{aligned}\Phi(s) &= \mathsf{tail}_{|s|}(s \mathbin{@} \mathsf{fB}(s * a_s))\\ &= a_s * \mathsf{tail}_{|s|+1}(\mathsf{fB}(s * a_s))\\ &\overset{(*)}{=} X_s * \mathsf{tail}_{|s|+1}(\mathsf{fB}(s * X_s))\\ &= X_s * \Phi(s * X_s)\end{aligned}$$

where for $(*)$ we have

$$\begin{aligned}a_s &= \phi^{\varepsilon,\Delta}_s(\lambda x \,.\, \mathsf{fB}(s * x))\\ &= \varepsilon_{|s|}(\lambda x \,.\, \Delta_{|s|}(\mathsf{fB}(s * x)))\\ &= \varepsilon_{|s|}(\lambda x \,.\, \Delta_{|s|}(s * x * \mathsf{tail}_{|s|+1}(\mathsf{fB}(s * x))))\\ &= \varepsilon_{|s|}(\lambda x \,.\, \Delta_{|s|}(s * x * \Phi(s * x)))\\ &= X_s.\end{aligned}$$

For the converse, define $\Phi^{\phi,n}(s) := s * \mathsf{sfB}^{\varepsilon^{\phi},\Delta}_{\rho,\rho^*}(s)$ where $\Delta_n \overset{\rho^*\to\rho^*}{:=} \mathrm{id}$ and

$$\varepsilon_n(p^{\rho\to\rho^*}) := \phi_{[p(0)](n)}(p).$$

Then we have $\mathsf{fB}_{\rho}[\Phi]$. If $|s| > n$ then $\Phi(s) = s * \langle\rangle$, and otherwise

$$\begin{aligned}\Phi(s) &= s * X_s * \mathsf{sfB}(s * X_s)\\ &\overset{(*)}{=} s * a_s * \mathsf{sfB}(s * a_s)\\ &= s \mathbin{@} \Phi(s * a_s)\end{aligned}$$

where for $(*)$ we have

$$\begin{aligned}X_s &= \varepsilon_{|s|}(\lambda x \,.\, s * x * \mathsf{sfB}(s * x))\\ &= \phi_{[s*x*\mathsf{sfB}(s*0)](|s|)}(\lambda x \,.\, s * x * \mathsf{sfB}(s * x))\\ &= \phi_s(\lambda x \,.\, \Phi(s * x))\end{aligned}$$

□

Characterising the terms of P_n

The recursor R_ρ can be viewed as an object that implements computations on a infinite array of type $\rho^{\mathbb{N}}$. It assigns a value $\mathsf{R}_\rho(0) := y$ to position 0, and proceeds to write values sequentially along the whole array, where $\mathsf{R}_\rho(n) := z_n(\mathsf{R}_\rho(0), \ldots, \mathsf{R}_\rho(n-1))$ is determined by the function z_n which can make its decision based on the values of (potentially all) previous positions on the array.

The finite bar recursor fB_ρ carries out computations along the same array $\rho^{\mathbb{N}}$. However, having already assigned values $\langle s_0, \ldots, s_{n-1}\rangle$ to the first n positions, the value at n is given by $\mathsf{fB}_\rho(s) := \phi_s(\lambda x \,.\, \mathsf{fB}(s*x))$. Not only does fB have access to previously computed values, but it can *test* the outcome of playing potential values x at $|s|$ courtesy of the function $\lambda x \,.\, \mathsf{fB}(s * x)$.

This suggests that fB_ρ is a stronger form of recursion than R_ρ. In the next section we make this precise by demonstrating that fB_ρ is capable of not only defining the primitive recursive functionals of level $\deg(\rho)$, but all primitive recursive functionals of level $\deg(\rho)+1$.

10.2 The equivalence of the fragments P_n and T_{n+1}.

Notation. Given two fragments X and Y of system T, we write $\mathsf{X} \Rightarrow \mathsf{Y}$ if all terms in Y can be defined in X.

It is clear that finite bar recursion is primitive recursively definable. More specifically:

Theorem 10.6. *The recursor* $\mathsf{R}_{\rho^* \to \rho^*}$ *defines* fB_ρ*, and therefore* $\mathsf{T}_{n+1} \Rightarrow \mathsf{P}_n$*.*

Proof. It is apparent from its defining equations that $\mathsf{fB}^{\phi,n}_\rho$ is just a standard recursion of type $\rho^* \to \rho^*$ in which the quantity[1] $n + 1 - |s|$ decreases until it reaches 0. Define

$$\begin{aligned} y &:= \lambda s \,.\, s \\ z^{\phi,n}_i(F^{\rho^* \to \rho^*}) &:= \lambda s \,.\, s \ @\ F(s * a_s) \end{aligned}$$

where $a_s := \phi_{[F(s)](n-i)}(\lambda x \,.\, F(s * x))$. Define $\Phi^{\phi,n}(s) := \mathsf{R}^{y,z^{\phi,n}}_{\rho^* \to \rho^*}(n + 1 - |s|)(s)$. Then $\mathsf{fB}_\rho[\Phi]$ holds. For $|s| > n$ we have

$$\Phi(s) = \mathsf{R}(0)(s) = y(s) = s,$$

and for $|s| \leq n$:

$$\begin{aligned} \Phi(s) &= \mathsf{R}(n + 1 - |s|)(s) \\ &= z_{n-|s|}(\mathsf{R}(n - |s|))(s) \\ &= s \ @\ \mathsf{R}(n - |s|)(s * a_s) \\ &= s \ @\ \Phi(s * a_s) \end{aligned}$$

where

$$\begin{aligned} a_s &= \phi_{[\mathsf{R}(n-|s|)(s)](n-n+|s|)}(\lambda x \,.\, \mathsf{R}(n - |s|)(s * x)) \\ &= \phi_s(\lambda x \,.\, \Phi(s * x)). \end{aligned}$$

□

Before we prove the converse to Theorem 10.6, let us first show how fB_ρ easily defines R_ρ without any kind of backtracking.

[1] By '$-$' we always mean truncated subtraction, with $n - i = 0$ for $i \geq n$.

Proposition 10.7. *The finite bar recursor* fB_ρ *defines* R_ρ*, and therefore* $\mathsf{P}_n \Rightarrow \mathsf{T}_n$.

Proof. Define $\phi_s^{y,z}$ as the constant function

$$\phi_s^{y,z}(p^{\rho\to\rho^*}) \overset{\rho}{:=} \begin{cases} y & \text{if } |s| = 0 \\ z_{|s|-1}(s_{|s|-1}) & \text{if } |s| > 0 \end{cases}$$

and let $\Phi^{y,z}(n){:=}\mathsf{fB}_\rho^{\phi^{y,z},n}(\langle\rangle)_n$. We have $\mathsf{R}_\rho[\Phi]$ because

$$\Phi(0) = \mathsf{fB}^{\phi,0}(\langle\rangle)_0 = \phi_{\langle\rangle}(\lambda x\ .\ \mathsf{fB}^{\phi,0}(\langle x\rangle)) = y$$

and for $n > 0$

$$\begin{aligned} \Phi(n+1) &= \mathsf{fB}^{\phi,n}(\langle\rangle)_n \\ &\overset{P.10.4}{=} \phi_{[\mathsf{fB}^{\phi,n}(\langle\rangle)](n)}(\lambda x\ .\ [\mathsf{fB}^{\phi,n}(\langle\rangle)](n) * x) \\ &= z_{n-1}(\mathsf{fB}^{\phi,n}(\langle\rangle)_{n-1}) \\ &\overset{(*)}{=} z_{n-1}(\mathsf{fB}^{\phi,n-1}(\langle\rangle)_{n-1}) \\ &= z_{n-1}(\Phi(n-1)). \end{aligned}$$

For (ϕ) we need to verify that $\mathsf{fB}^{\phi,n}(\langle\rangle)_{n-1} = \mathsf{fB}^{\phi,n-1}(\langle\rangle)_{n-1}$, which is easily done by induction on $\mathsf{fB}^{\phi,n}(\langle\rangle)_i = \mathsf{fB}^{\phi,n-1}(\langle\rangle)_i$. □

Already we get the following key result.

Corollary 10.8. *Gödel's system* T *can be equivalently defined by using the finite bar recursors* fB *instead of the primitive recursors* R.

However, if we appeal to the full power of fB and allow backtracking, we strengthen Proposition 10.7.

Theorem 10.9. *The finite bar recursor* fB_ρ *defines* $\mathsf{R}_{\rho\to\rho}$.

Proof. Given $y\colon \rho \to \rho$, $z\colon \mathbb{N} \to (\rho \to \rho)$ and an argument $a\colon \rho$, define $\phi_s^{n,a,y,z}$ by

$$\phi_s^{n,a,y,z}(p^{\rho\to\rho^*}) \overset{\rho}{:=} \begin{cases} z_{n-1}(\lambda x\ .\ p(x)_1)(a) & \text{if } |s| = 0 \\ z_{n-|s|-1}(\lambda x\ .\ p(x)_{|s|+1})(|s|_{|s|-1}) & \text{if } 0 < |s| < n \\ y(s_{|s|-1}) & \text{if } |s| = n \\ 0_\rho & \text{otherwise} \end{cases}$$

for $n > 0$, and

$$\phi_s^{0,a,y,z}(p^{\rho\to\rho^*}) \overset{\rho}{:=} \begin{cases} y(p(a)_0) & \text{if } |s| = 0 \\ 0_\rho & \text{otherwise.} \end{cases}$$

Note that technically for the ε to be well defined we should add

$$p(x)_i := p(x) \text{ if } i < |p(x)|, \text{ else } 0_\rho.$$

Now, define $\Phi^{y,z}(n) \overset{\rho\to\rho}{:=} \lambda a \,.\, \mathsf{fB}^{\phi^{n,a,y,z},n}(\langle\rangle)_0$. Then we have $\mathsf{R}_{\rho\to\rho}[\Phi]$. For $n = 0$

$$\Phi(0)(a) = \mathsf{fB}^{\phi^{0,a},0}(\langle\rangle) \overset{P.10.4}{=} \phi^{0,a}_{\langle\rangle}(\lambda x \,.\, \mathsf{fB}^{\phi^{0,a},0}(\langle x\rangle)) = y(\mathsf{fB}^{\phi^{0,a},0}(\langle a\rangle)_0) = y(a)$$

and therefore $\Phi(0) = y$. For $n > 0$ we have

$$\begin{aligned}
\Phi(n)(a) &= \mathsf{fB}^{\phi^{n,a},n}(\langle\rangle)_0 \\
&= \phi^{n,a}_{\langle\rangle}(\lambda x \,.\, \mathsf{fB}^{\phi^{n,a},n}(\langle x\rangle)) \\
&= z_{n-1}(\lambda x \,.\, \mathsf{fB}^{\phi^{n,a},n}(\langle x\rangle)_1)(a) \\
&\overset{(*)}{=} z_{n-1}(\lambda x \,.\, \mathsf{fB}^{\phi^{n-1,x},n-1}(\langle\rangle)_0)(a) \\
&= z_{n-1}(\Phi(n-1))(a)
\end{aligned}$$

and therefore $\Phi(n) = z_{n-1}(\Phi(n-1))$. The main content of the proof is the verification of $(*)$, in which we carry out finite bar induction on the formula

$$P(s) :\equiv \mathsf{fB}^{\phi^{n,a},n}(\langle x\rangle * s)_{|s|+1} = \mathsf{fB}^{\phi^{n-1,x},n-1}(s)_{|s|}.$$

Then λ-abstracting from $P(\langle\rangle)$ we obtain

$$(*) \quad \lambda x \,.\, \mathsf{fB}^{\phi^{n,a},n}(\langle x\rangle)_1 = \lambda x \,.\, \mathsf{fB}^{\phi^{n-1,x},n-1}(\langle\rangle)_0.$$

First note that by Proposition 10.4, for $|s| \leq n-1$ the formula $P(s)$ is equivalent to

$$\phi^{n,a}_{\langle x\rangle * s}(\lambda x' \,.\, \mathsf{fB}^{\phi^{n,a},n}(\langle x\rangle * s * x')) = \phi^{n-1,a}_{s}(\lambda x' \,.\, \mathsf{fB}^{\phi^{n-1,x},n-1}(s * x'))$$

Now, inspecting the definition of ϕ_s we see $P(s)$ clearly holds for $|s| = n-1$ by

$$y((\langle x\rangle * s)_{n-1}) = y(s_{n-2}).$$

To establish the bar induction step there are two cases to deal with. For $0 < |s| < n-1$ $P(s)$ follows by

$$\begin{aligned}
&z_{n-|\langle x\rangle * s|-1}(\lambda x' \,.\, \mathsf{fB}^{\phi^{n,a},n}(\langle x\rangle * s * x')_{|\langle x\rangle * s|+1})(((\langle x\rangle * s)_{|\langle x\rangle * s|-1}) \\
&\overset{B.I.H.}{=} z_{n-|s|-2}(\lambda x' \,.\, \mathsf{fB}^{\phi^{n-1,x},n-1}(s * x')_{|s|+1})(s_{|s|-1})
\end{aligned}$$

and finally for $|s| = 0$ by

$$z_{n-2}(\lambda x' \,.\, \mathsf{fB}^{\phi^{n,a},n}(\langle x\rangle * x')_2)(\langle x\rangle_0) \overset{B.I.H.}{=} z_{n-2}(\lambda x' \,.\, \mathsf{fB}^{\phi^{n-1,x},n-1}(x')_1)(x)$$

which completes the proof. $\square$

Corollary 10.10. $\mathsf{P}_n \Leftrightarrow \mathsf{T}_{n+1}$

Proof. This follows immediately from Theorems 10.6 and 10.9. For the direction $\mathsf{P}_n \Rightarrow \mathsf{T}_{n+1}$ note that using the product type any type σ of degree $n+1$ is equivalent to one of the form $\rho \to \mathbb{N}$ where $\deg(\rho) = n$, which can clearly be encoded in the type $\rho \to \rho$. Therefore R_σ can be defined from $\mathsf{R}_{\rho\to\rho}$ and hence from fB_ρ. □

A consequence of Theorem 10.9 is that the Ackermann function can be defined from finite bar recursion of lowest type, and moreover by inspection we can construct it explicitly in terms of $\mathsf{fB}_\mathbb{N}$.

Example 10.11 (Ackermann function in P_0). Given natural numbers a and n, define $\phi_s^{n,a}$ by

$$\phi_s^{n,a}(p^{\mathbb{N}\to\mathbb{N}^*}) \overset{\mathbb{N}}{:=} \begin{cases} (\lambda x \,.\, p(x)_1)^{(a+1)}(1) & \text{if } |s| = 0 \\ (\lambda x \,.\, p(x)_{|s|+1})^{(s_{|s|-1}+1)}(1) & \text{if } 0 < |s| < n \\ s_{|s|-1} + 1 & \text{if } |s| = n \\ 0 & \text{otherwise} \end{cases}$$

for $n > 0$, and

$$\phi_s^{0,a}(p) \overset{\mathbb{N}}{:=} \begin{cases} a+1 & \text{if } |s| = 0 \\ 0 & \text{otherwise,} \end{cases}$$

where $f^{(i)}$ is defined in T_0 as usual by $f^{(0)}(x) = x$ and $f^{(i+1)}(x) = f(f^{(i)}(x))$. Then by Theorem 10.9,

$$A(n,a) \overset{\mathbb{N}}{:=} \mathsf{fB}_\mathbb{N}^{\phi^{n,a},n}(\langle\rangle)_0$$

satisfies the defining equations of the Ackermann function (cf. Example 2.7). To see this, note that (using $(*)$ from the proof of Theorem 10.9)

$$\begin{aligned}
A(0,a) &= \varphi^{0,a}_{\langle\rangle}(\lambda x\ .\ \mathsf{fB}^{\phi^{0,a},0}(\langle x\rangle)) \\
&= a+1; \\
A(n,0) &= \phi^{n,0}_{\langle\rangle}(\lambda x\ .\ \mathsf{fB}^{\phi^{n,0},n}(\langle x\rangle)) \\
&= (\lambda x\ .\ \mathsf{fB}^{\phi^{n,0},n}(\langle x\rangle)_1)(1) \\
&= \mathsf{fB}^{\phi^{n,0},n}(\langle 1\rangle)_1 \\
&\overset{(*)}{=} \mathsf{fB}^{\phi^{n-1,1},n-1}(\langle\rangle)_0 \\
&= A(n-1,1) \\
A(n,a) &= \phi^{n,a}_{\langle\rangle}(\lambda x\ .\ \mathsf{fB}^{\phi^{n,a},n}(\langle x\rangle)) \\
&= (\lambda x\ .\ \mathsf{fB}^{\phi^{n,a},n}(\langle x\rangle)_1)^{(a+1)}(1) \\
&= \mathsf{fB}^{\phi^{n,a},n}(\langle(\lambda x\ .\ \mathsf{fB}^{\phi^{n,a},n}(\langle x\rangle)_1)^{(a)}(1)\rangle)_1 \\
&\overset{(*)}{=} \mathsf{fB}^{\phi^{n-1,b},n-1}(\langle\rangle)_0 \\
&= A(n-1,b)
\end{aligned}$$

where $b = (\lambda x\ .\ \mathsf{fB}^{\phi^{n,a},n}(\langle x\rangle)_1)^{(a)}(1)$. But by $(*)$ we have

$$\mathsf{fB}^{\phi^{n,a},n}(\langle x\rangle)_1 = \mathsf{fB}^{\phi^{n-1,x},n-1}(\langle\rangle)_0 = \mathsf{fB}^{\phi^{n,a-1},n}(\langle x\rangle)_1$$

and therefore

$$\begin{aligned}
b &= (\lambda x\ .\ \mathsf{fB}^{\phi^{n,a-1},n}(\langle x\rangle)_1)^{(a)}(1) \\
&= \phi^{n,a-1}_{\langle\rangle}(\lambda x\ .\ \mathsf{fB}^{\phi^{n,a-1},n}(\langle x\rangle)) \\
&= \mathsf{fB}^{\phi^{n,a-1},n}(\langle\rangle)_0 \\
&= A(n,a-1)
\end{aligned}$$

This construction of the Ackermann function using $\mathsf{fB}_{\mathbb{N}}$ gives us an intuitive way of visualising the function as a backtracking procedure over sequences of natural numbers. For example, we sketch the computation of $A(3,a)$ using $\mathsf{fB}^{\phi^{3,a},3}$ below:

$$\begin{aligned}
&\mathsf{fB}(\langle x_0,x_1,x_2\rangle)_3 = \phi_{\langle x_0,x_1,x_2\rangle}(\lambda x\ .\ \langle x_0,x_1,x_2,x\rangle_3) = x_2+1; \\
&\mathsf{fB}(\langle x_0,x_1\rangle)_2 = \phi_{\langle x_0,x_1\rangle}(\lambda x\ .\ \mathsf{fB}(\langle x_0,x_1,x\rangle)_2) = (\lambda x\ .\ x+1)^{(x_1+1)}(1) = x_1+2; \\
&\mathsf{fB}(\langle x_0\rangle)_1 = \phi_{\langle x_0\rangle}(\lambda x\ .\ \mathsf{fB}(\langle x_0,x\rangle)_2) = (\lambda x\ .\ x+2)^{x_0+1}(1) = 2x_0+3 \\
&\mathsf{fB}(\langle\rangle)_0 = \phi_{\langle\rangle}(\lambda x\ .\ \mathsf{fB}(\langle x\rangle)_1) = (\lambda x\ .\ 2x+3)^{(a+1)}(1) = 2^{(a+3)}-3.
\end{aligned}$$

10.3 Interpreting fragments of arithmetic.

We conclude by calibrating which fragments of finite bar recursion are necessary for interpreting fragments of arithmetic based on bounded collection, or equivalently finite choice.

In fact the latter is most naturally interpreted using finite bar recursion, so we focus purely on this principle (the functional interpretation of bounded collection using finite bar recursion is discussed in e.g. [31]). We begin with a standard result.

Lemma 10.12. *A formula A in the language of Peano arithmetic* PA *has ND interpretation $|A^{\mathrm{N}}|^x_y$ where x and y are (tuples of) variables in the language of all finite types. For $n > 1$:*

(a) If A is a Π^0_n formula then the tuple x contains variables of type degree at most $n-1$ and the tuple y contains variables of type degree at most $n-2$;

(b) If A is a Σ^0_n formula then the tuple x contains variables of type degree at most n and the tuple y contains variables of degree at most $n-1$.

Proof. Straightforward induction on n, although we always eliminate unnecessary negations. In the case that A is Π^0_n it is clear that $A^{\mathrm{N}} \leftrightarrow A^*$ using the intuitionistic law $\neg\neg\forall x \neg\neg A(x) \leftrightarrow \forall x \neg\neg A(x)$. □

C. Parsons uses this lemma to extend Gödel's functional interpretation to the induction based fragments of arithmetic.

Theorem 10.13 (Parsons [73]). *The functional interpretation of* Π_n-IND *requires only the recursor* R_ρ *of type level* $\deg(\rho) = n-1$*. Therefore* $\mathsf{I}\Pi_n$ *(or equivalently* $\mathsf{I}\Sigma_n$*) has a functional interpretation is* T_{n-1}*.*

We now provide an analogue of Parsons' result for the principle of finite choice.

Definition 10.14. Let $\tilde{\mathsf{P}}_n$ be the subset of P_n containing the finite product $\mathsf{fP}_{\rho,\tau}$ restricted to simple selection functions (i.e. those with $\varepsilon_s = \varepsilon_{|s|}$ only dependent on the *length* of s) and with $\deg(\rho) \leq n$ and $\deg(\tau) \leq n-1$.

Remark 10.15. Note that it is not clear whether fB_ρ for $\deg(\rho) = n$ lies in $\tilde{\mathsf{P}}_n$ as its construction in Proposition 10.3 depends on $\mathsf{fP}_{\rho,\rho^*}$.

Theorem 10.16. *(a) The functional interpretation of* Π_1-FAC *requires only the finite product of selection functions* $\mathsf{fP}_{\mathbb{N},\mathbb{N}}$ *of lowest type. Therefore* $\mathsf{F}\Pi_1$ *(or equivalently* $\mathsf{F}\Sigma_2$, $\mathsf{B}\Pi_1$ *etc.) has a functional interpretation in* P_0*.*

(b) The functional interpretation of Π_n-FAC *for $n > 1$ requires only the finite product of simple selection functions* $\mathsf{fP}_{\rho,\tau}$ *of type level* $\deg(\rho) = n-1$ *and* $\deg(\tau) = n-2$*. Therefore* $\mathsf{F}\Pi_n$ *(or equivalently* $\mathsf{F}\Sigma_{n+1}$, $\mathsf{B}\Pi_n$ *etc.) has a functional interpretation in* $\tilde{\mathsf{P}}_{n-1} \subseteq \mathsf{P}_{n-1}$*.*

Proof. The negative translation of finite choice

$$\forall i \leq m \exists x^{\mathbb{N}} A_i(x) \to \exists s^{\mathbb{N}^*} \forall i \leq m A_i(s_i)$$

for a Π^0_n formula $A_i(x)$ is equivalent to the formula

$$\forall i \leq m \neg\neg\exists x A_i(x)^{\mathrm{N}} \to \neg\neg\exists s \forall i \leq m A_i(s_i)^{\mathrm{N}},$$

which is partially interpreted as

$$\forall i \leq m \neg\neg\exists x^{\mathbb{N}}, \tilde{x}^{\rho} \forall \tilde{y}^{\tau} |A_i(x)^{\mathrm{N}}|^{\tilde{x}}_{\tilde{y}} \to \neg\neg\exists s^{\mathbb{N}^*}, \tilde{s}^{\tilde{\rho}^*} \forall i \leq m, \tilde{y} |A_i(s_i)^{\mathrm{N}}|^{\tilde{s}_i}_{\tilde{y}}$$

Relabelling variables and setting $B_i(x,y) := |A_i(x_0)^{\mathrm{N}}|^{x_1}_{y}$ this is equivalent to

$$\forall i \leq m \neg\neg\exists x^{\rho} \forall y^{\tau} B_i(x,y) \to \neg\neg\exists s^{\rho^*} \forall i \leq m, y B_i(s_i, y)$$

where (assuming $n > 1$) by Lemma 10.12, $\deg(\rho) = n-1$ and $\deg(\tau) = n-2$. But this has functional interpretation equivalent to

$$\forall \varepsilon, q, \varphi \exists s, p, i((i \leq m \to B_i(\varepsilon_i p, p(\varepsilon_i p))) \to (\varphi(s) \leq m \to B_{\varphi(s)}(s_{\varphi(s)}, q(s)))).$$

which can be solved by $s := \mathsf{fP}^{\varepsilon,q,m}_{\rho,\tau}(\langle\rangle)$, $i := \varphi(s)$ and $p := \lambda x \,.\, q(\mathsf{fP}^{\varepsilon,q}_{\rho,\tau,m}([s](i) * x))$, since by Proposition 10.4 for $\varphi(s) \leq m$ we have $s_{\varphi(s)} = \varepsilon_i(p)$ and $p(\varepsilon_i p) = q(s)$.

For the case $n = 1$ we have $\exists x^{\mathbb{N}} A_i(x)^{\mathrm{N}} \equiv \exists x^{\mathbb{N}} \forall y^{\mathbb{N}} A_i(x,y)$, and therefore by an analogous argument the functional interpretation of Π_1-FAC can be solved by the finite product $\mathsf{fP}_{\mathbb{N},\mathbb{N}}$ of lowest type. □

We sketch the results of this Chapter (in conjunction with those in [71, 72, 73]) in Figure 10.1.

$$\begin{tikzcd}
\mathsf{I}\Sigma_1 \arrow[dd] & \mathsf{F}\Pi_1 \arrow[l] \arrow[dd] & \mathsf{I}\Sigma_n \arrow[l, dotted] \arrow[dd] & \mathsf{F}\Pi_n \arrow[l] \arrow[d, "\text{Th.10.16}"] & \mathsf{I}\Sigma_{n+1} \arrow[l] \arrow[dd] & \mathsf{PA} \arrow[l, dotted] \arrow[dd] \\
 & & & \tilde{\mathsf{P}}_{n-1} \arrow[d, Rightarrow] & & \\
\mathsf{T}_0 & \mathsf{P}_0 \arrow[l, Rightarrow] & \mathsf{T}_{n-1} \arrow[l, dotted] & \mathsf{P}_{n-1} \arrow[l, Rightarrow, "\text{Prop. 10.7}"'] \arrow[r, Leftrightarrow, "\text{Th. 10.6, 10.9}"'] & \mathsf{T}_n & \mathsf{T} \arrow[l, dotted]
\end{tikzcd}$$

Figure 10.1: Fragments of Peano arithmetic and corresponding fragments of system T.

Remark 10.17. Our work in this section is nowhere near as extensive as Parsons' work on the induction fragments [72, 73]. Here, among other things, a more refined result is given for the functional interpretation of fragments based on the induction *rule*, and also a proof that the interpretation of $\mathsf{I}\Sigma_n$ in T_{n-1} is faithful, from which we can calibrate precisely the provably recursive functions of $\mathsf{I}\Sigma_n$ as those of T_{n-1}.

Remark 10.18. We leave open the question of whether $\tilde{\mathsf{P}}_{n-1}$ is weaker than P_{n-1} (and hence T_n). If not, then it is still possible that $\mathsf{F}\Pi_n$ could be interpreted in a weaker fragment of T. For the case of $\mathsf{F}\Pi_1$ it is completely open whether or not Π_1-FAC can be interpreted in some system F strictly between T_0 and T_1.

Remark 10.19. Finally, we observe that Theorem 10.16 can be easily adapted to show that the full, dependent product of selection functions interprets the following finite analogue of dependent choice:

$$\mathsf{FDC}\ :\ \forall i \leq m, x \exists y A_i(x,y) \to \exists s \forall i \leq m A_i(s_i, s_{i+1}).$$

Kreuzer [55, Chapter 4.2] points out that FDC for Π^0_1 formulas is, unlike Π_1-FAC, equivalent to Σ_2-IND, which implies that $\mathsf{fP}_{\mathbb{N},\mathbb{N}}$ (or equivalently $\mathsf{fB}_{\mathbb{N},\mathbb{N}}$) defines all provably recursive functions of $\mathsf{I}\Sigma_2$. In particular this yields an alternative proof that $\mathsf{fB}_{\mathbb{N},\mathbb{N}}$ defines the Ackermann function, although of course we are able to give an explicit construction here.

CHAPTER

ELEVEN

On Spector's 'Special' and 'General' Forms of Bar Recursion

In his original paper on the computational interpretation of analysis [86], Spector introduces a general scheme of bar recursion GBR but draws attention to the fact that only a 'special' form of bar recursion SBR is required to interpret countable choice (cf. Chapter 4).

It is clear that SBR is definable from GBR. In this short chapter we prove the converse, namely that SBR is actually strong enough to primitive recursively define GBR over system T. This result follows firstly from the fact that the Kreisel/Howard trick for defining Spector's search functional μ_{Sp} using GBR (Proposition 4.1) can in fact be carried out using SBR, and secondly from the result that a bound for the search functional when it exists can be defined in system T (Proposition 4.2). This allows us to define an intermediate form of 'finite but unbounded' bar recursion FBR, which is capable of defining full bar recursion GBR.

11.1 The Kreisel/Howard trick via SBR

The first part of the proof involves adapting the Kreisel/Howard construction of Spector's search functional μ_{Sp} using SBR. This uses ideas from the previous chapter, in which bar recursion was viewed as a device that carries out backtracking computations along an array of type $\rho^{\mathbb{N}}$ (see Section 10.1).

Proposition 11.1. *For any type ρ there exists a term t of $\mathsf{E\text{-}HA}^{\omega} + \mathsf{SBR}$ such that*

$$\exists n \leq t_{\varphi,\alpha}(\varphi(\overline{\alpha, n}) < n) \tag{11.1}$$

for any $\varphi \colon \rho^{\mathbb{N}} \to \mathbb{N}$, $\alpha \colon \rho^{\mathbb{N}}$. Therefore Spector's search term $\mu_{Sp}(\varphi, \alpha)$ that finds the least n satisfying Spector's condition can be defined in $\mathsf{E\text{-}HA}^{\omega} + \mathsf{SBR}$.

Proof. Without loss of generality assume that $\rho = \sigma \to \mathbb{N}$ and let n_ρ denote the constant functional $\lambda x^\sigma \,.\, n$ of type ρ (and just the natural number n if $\rho = \mathbb{N}$). Furthermore, given $f\colon \rho$ let $n_\rho + f$ denote the functional $\lambda x^\sigma \,.\, n + fx$. Define the sequence of selection functions $\phi^{\varphi,\alpha}\colon \mathbb{N} \to J_{\rho^{\mathbb{N}}}\rho$ by

$$\phi_i^{\varphi,\alpha}(p^{\rho\to\rho^{\mathbb{N}}}) \overset{\rho}{:=} \begin{cases} 0_\rho & \text{if } \exists n \leq i(\varphi(\overline{\alpha,n}) < n) \\ 1_\rho + p(\alpha i)_{i+1} & \text{otherwise} \end{cases}$$

and let $\beta(i) \overset{\mathbb{N}}{:=} \mathsf{SBR}_\rho^{\phi^{\varphi,\alpha},\mathrm{id},\varphi}([\alpha](i))_i(0_\sigma)$ where $\rho = \sigma \to \mathbb{N}$ (or just $\beta(i) \overset{\mathbb{N}}{:=} \mathsf{SBR}_\rho^{\phi^{\varphi,\alpha},\mathrm{id},\varphi}([\alpha](i))_i$ if $\rho = \mathbb{N}$). We claim that β satisfies

$$\beta i = \begin{cases} 0 & \text{if } \exists n \leq i(\varphi(\overline{\alpha,n}) < n) \\ 1 + \beta(i+1) & \text{otherwise,} \end{cases}$$

then from this we are able to show that $\exists n \leq \beta(0)(\varphi(\overline{\alpha,n}) < n)$ as in the proof of Proposition 4.1, and so $t := \beta(0)$ satisfies (11.1). It remains to prove the claim.

If $\exists n \leq i(\varphi(\overline{\alpha,n}) < n)$ then either $\varphi(\overline{\alpha,i}) < i$, in which case

$$\beta i = \mathsf{EPS}([\alpha](i))_i(0_\sigma) = 0_\rho(0_\sigma) = 0$$

or $\varphi(\overline{\alpha,i}) \geq i$ and $\exists n < i(\varphi(\overline{\alpha,n}) < n)$, in which case

$$\beta i = \mathsf{SBR}([\alpha](i))_i(0_\sigma) = \phi_i(\lambda x \,.\, \mathsf{SBR}([\alpha](i) * x))(0_\sigma) = 0_\rho(0_\sigma) = 0.$$

On the other hand, if $\forall n \leq i(\varphi(\overline{\alpha,n}) \geq n)$ we must have

$$\begin{aligned} \beta i &= \mathsf{SBR}([\alpha](i))_i(0)\sigma) \\ &= \phi_i(\lambda x \,.\, \mathsf{SBR}([\alpha](i) * x))(0_\sigma) \\ &= (1_\rho + \mathsf{SBR}([\alpha](i+1))_{i+1})(0_\sigma) \\ &= 1 + \mathsf{SBR}([\alpha](i+1))_{i+1}(0_\sigma) \\ &= 1 + \beta(i+1). \end{aligned}$$

□

Corollary 11.2. *For each k there is a term $\tilde{\mu}^k_{Sp}$ of* $\mathsf{E\text{-}HA}^\omega$ *such that, provably in* $\mathsf{E\text{-}HA}^\omega + \mathsf{SBR}$*, $\tilde{\mu}^k_{Sp}(\varphi,\alpha)$ is the least integer $N > k$ satisfying Spector's condition $\varphi(\overline{\alpha,N}) < N$.*

Proof. Define $\tilde{\mu}^0_{\mathrm{Sp}} := \tilde{\mu}_{\mathrm{Sp}}$ as in Proposition 4.2. Then analogous to the proof of Proposition 4.2, but this time using Proposition 11.1, we can show that

$$\mathsf{E\text{-}HA}^\omega + \mathsf{SBR} \vdash \varphi(\overline{\alpha,\tilde{\mu}^0_{\mathrm{Sp}}(\varphi,\alpha)}) < \tilde{\mu}^0_{\mathrm{Sp}}(\varphi,\alpha) \wedge (n < \tilde{\mu}^0_{\mathrm{Sp}}(\varphi,\alpha) \to \varphi(\overline{\alpha,n}) \geq n).$$

It is easy to see that $\tilde{\mu}^k_{\mathrm{Sp}}$ is definable from $\tilde{\mu}^0_{\mathrm{Sp}}$ for any $k > 0$. Simply set

$$\tilde{\mu}^k_{\mathrm{Sp}}(\varphi,\alpha) := \tilde{\mu}^0_{\mathrm{Sp}}(\underbrace{\lambda\beta \,.\, \varphi([\alpha](k) * \beta) - k}_{\tilde{\varphi}}, \mathsf{tail}_k(\alpha))$$

where we recall that tail_k removes the first k entries of α. Then noting that

$$\tilde{\varphi}(\overline{\mathsf{tail}_k(\alpha), N}) = \varphi([\alpha](k) * \overline{\mathsf{tail}_k(\alpha), N}) - k = \varphi(\overline{\alpha, k+N}) - k$$

we see that the least N satisfying $\tilde{\varphi}(\overline{\mathsf{tail}_k(\alpha), N}) < N$ is also the least satisfying $\varphi(\overline{\alpha, k+N}) < k+N$. □

11.2 Defining GBR from SBR

The primitive recursive definability of μ_{Sp} allows us to devise a variant of Spector's special bar recursion which embodies the stopping condition: returning a finite instead of infinite sequence. The constant FBR_ρ has defining equation

$$\mathsf{FBR}^{\phi,\varphi}_{\rho}(s) \overset{\rho^*}{:=} s \mathbin{@} \begin{cases} \langle\rangle & \text{if } \varphi(\hat{s}) < |s| \\ \mathsf{FBR}^{\phi,\varphi}(s * a_s) & \text{otherwise} \end{cases}$$

for $a_s := \phi_s(\lambda x \,.\, \mathsf{FBR}(s * x))$, where $\phi\colon \rho^* \to J_{\rho^*}\rho$ and $\varphi\colon \rho^{\mathbb{N}} \to \mathbb{N}$.

Lemma 11.3. SBR *primitive recursively defines* FBR.

Proof. Given parameters ϕ, φ for FBR, define $\tilde{\phi}\colon \rho^* \to J_{\rho^{\mathbb{N}}}\rho$ by

$$\tilde{\phi}_s(p^{\rho \to \rho^{\mathbb{N}}}) \overset{\rho}{:=} \phi_s(\lambda x \,.\, [p(x)](N_{p(x),|s|}))$$

where $N_{\alpha,k} := \tilde{\mu}^{k-1}_{\mathrm{Sp}}(\varphi, \alpha)$ i.e. $N_{\alpha,k} \geq k$ is the least satisfying $\varphi(\overline{\alpha, N_{\alpha,k}}) < N_{\alpha,k}$ (provably in $\mathsf{E\text{-}HA}^\omega + \mathsf{SBR}$). Then the term $\Phi^{\phi,\varphi}(s) \overset{\rho^*}{:=} [\mathsf{SBR}^{\tilde{\phi},\varphi}(s)](N_{\mathsf{SBR}^{\tilde{\phi},\varphi}(s),|s|})$ satisfies $\mathsf{FBR}_\rho[\Phi]$. For $\varphi(\hat{s}) < |s|$ we have $N_{\mathsf{SBR}(s),|s|} = |s|$ and therefore

$$\Phi(s) = [\mathsf{SBR}(s)](|s|) = s.$$

On the other hand, if $\varphi(\hat{s}) > |s|$ then $N_{\mathsf{SBR}(s),|s|} = N_{\mathsf{SBR}(s),|s|+1}$ and

$$\begin{aligned} \Phi(s) &= [\mathsf{SBR}(s)](N_{\mathsf{SBR}(s),|s|+1}) \\ &= [\mathsf{SBR}(s * \tilde{a}_s)](N_{\mathsf{SBR}(s*\tilde{a}_s),|s|+1}) \\ &\overset{(*)}{=} [\mathsf{SBR}(s * a_s)](N_{\mathsf{SBR}(s*a_s),|s|+1}) \\ &= \Phi(s * a_s) \end{aligned}$$

where for $(*)$ we have

$$\begin{aligned} \tilde{a}_s &= \tilde{\phi}_s(\lambda x \,.\, \mathsf{SBR}(s * x)) \\ &= \phi_s(\lambda x \,.\, [\mathsf{SBR}(s * x)](N_{\mathsf{SBR}(s*x),|s|})) \\ &= \phi_s(\lambda x \,.\, [\mathsf{SBR}(s * x)](N_{\mathsf{SBR}(s*x),|s|+1})) \\ &= \phi_s(\lambda x \,.\, \Phi(s * x)) \\ &= a_s. \end{aligned}$$

□

We now have everything we need to prove our main theorem. The intuition behind the proof is as follows. Spector's general bar recursion assigns a value $V_s := Y(s)$ to the nodes of the tree defined by Spector's condition, and computes the value $Z_s(\lambda x \,.\, V_{s*x})$ at intermediate nodes by querying the value of its children nodes. The special form of bar recursion involves the particular case where the value V_s is an infinite path extending s. We simulate GBR using FBR by encoding the value V_s as the first element in the path extending s. While this tactic would have been impossible using SBR, we can search for this value using FBR because here path extensions are always finite.

Theorem 11.4. *A single instance of* FBR *primitive recursively defines* GBR.

Notation. We need to extend $\mathsf{E\text{-}HA}^\omega$ to contain sum types $\rho+\tau$ (which are easily definable from the usual types). Given $a\colon \rho$ and $b\colon \tau$ define $[a]_\rho\colon \rho+\tau$, $[b]_\tau\colon \rho+\tau$ to be the standard injections into $\rho+\tau$, and assume we have predicates $x \in \rho$ and $x \in \tau$ that allow us to decide which type $x\colon \rho+\tau$ belongs to. Finally:

- Let $\check{s}\colon (\rho+\tau)^*$ denote the embedding of $s\colon \rho^*$ in $\rho+\tau$;
- Given $\alpha\colon (\rho+\tau)^{\mathbb{N}}$, define $\tilde{\alpha}\colon \rho^{\mathbb{N}}$ as

$$\tilde{\alpha}(i) \overset{\rho}{:=} \begin{cases} 0_\rho & \text{if } \alpha(i) \in \tau \\ \alpha(i) & \text{if } \alpha(i) \in \rho. \end{cases}$$

  Define $\tilde{s}\colon \rho^*$ for $s\colon (\rho+\tau)^*$ similarly

Proof of Theorem 11.4. Given parameters $Z\colon \rho^* \to ((\rho \to \tau) \to \tau)$, $Y\colon \rho^* \to \tau$ and $\varphi\colon \rho^{\mathbb{N}} \to \mathbb{N}$, primitive recursively define $\phi\colon (\rho+\tau)^* \to J_{(\rho+\tau)^*}(\rho+\tau)$, $q\colon (\rho+\tau)^* \to \tau$ and $\tilde{\varphi}\colon (\rho+\tau)^{\mathbb{N}} \to \mathbb{N}$ by

$$\phi_s(p^{(\rho+\tau)\to(\rho+\tau)^*}) \overset{\rho+\tau}{:=} [Z_{\tilde{s}}(\lambda x^\rho \,.\, q(p([x]_\rho)))]_\tau$$

$$q(s) \overset{\tau}{:=} \begin{cases} Y(\tilde{s}) & \text{if } \forall i < |s|(s_i \in \rho) \\ s_i & \text{for least } i < |s| \text{ such that } s_i \in \tau. \end{cases}$$

$$\tilde{\varphi}(\alpha) \overset{\tau}{:=} \varphi(\tilde{\alpha})$$

We claim that $\Phi(s) \overset{\tau}{:=} q(\mathsf{FBR}^{\phi,q}_{\rho+\tau}(\check{s}))$ satisfies $\mathsf{GBR}^{Z,Y,\varphi}_{\rho,\tau}[\Phi]$. Noting that

$$(\dagger) \quad \varphi(\hat{s}) < |s| \leftrightarrow \tilde{\varphi}(\hat{\check{s}}) < |\check{s}|,$$

for $\varphi(\hat{s}) < |s|$ we have

$$\Phi(s) = q(\check{s}) = Y(s),$$

and for $\varphi(\hat{s}) \geq |s|$

$$\begin{aligned}\Phi(s) &= q(\check{s} * [a_{\check{s}}]_\tau \mathbin{@} \mathsf{FBR}(\check{s} * a_{\check{s}})) \\ &= a_{\check{s}} \\ &= \phi_{\check{s}}(\lambda x^{\rho+\tau} \,.\, \mathsf{FBR}(s * x)) \\ &= Z_s(\lambda x^\rho \,.\, q(\mathsf{FBR}(\check{s} * [x]_\rho))) \\ &= Z_s(\lambda x^\rho \,.\, q(\mathsf{FBR}(s \mathbin{\check{*}} x))) \\ &= Z_s(\lambda x^\rho \,.\, \Phi(s * x)).\end{aligned}$$

□

Corollary 11.5. GBR*,* FBR *and* SBR *are all primitive recursively equivalent.*

Remark 11.6. The functional FBR is of interest in its own right as it forms a powerful equivalent formulation of the usual SBR in that it embodies Spector's search functional μ_{Sp}, which can be defined as

$$\mu_{\mathrm{Sp}}(\varphi, \alpha) := |\mathsf{FBR}_\rho^{\varepsilon^\alpha, \varphi}(\langle\rangle)|$$

for $\varepsilon_s(p) := p(\alpha(|s|))$.

Remark 11.7. While we have shown that Spector's 'special' bar recursion is in fact as general as his 'general' bar recursion in terms of T-definability, Spector was nevertheless correct to single out the fact that his solution to the functional interpretation of the double negation shift only makes use of SBR. In particular we know that $\mathsf{SBR}_{\mathbb{N}}$ (which is definable from $\mathsf{GBR}_{\mathbb{N},\mathbb{N}^\mathbb{N}}$) suffices to solve the functional interpretation of arithmetical comprehension, and despite our equivalence result this still does not follow from the fact that $\mathsf{GBR}_{\mathbb{N},\mathbb{N}^\mathbb{N}}$ also suffices to interpret arithmetic comprehension. This is because to define $\mathsf{GBR}_{\mathbb{N},\mathbb{N}^\mathbb{N}}$ we need $\mathsf{SBR}_{\mathbb{N}^\mathbb{N}}$, which is stronger than $\mathsf{SBR}_{\mathbb{N}}$ and in fact also capable of defining $\mathsf{GBR}_{\mathbb{N}^\mathbb{N},\mathbb{N}^\mathbb{N}}$ (see [70, Remark 5.4]). This point demonstrates quite nicely that while primitive recursive definability is a non-trivial relation between modes of recursion in all finite types, it leaves completely open the relative strength of equivalent modes of recursion at the level of types.

CHAPTER

TWELVE

Implicitly Controlled Bar Recursion and Open Recursion

The final chapter of this dissertation contains the main results of Part III, where we prove that modified bar recursion, the implicitly controlled product of selection functions, update recursion, open recursion and the BBC functional are all primitive recursively equivalent.

The equivalence of IPS and MBR (along with several other definability results) is established in [28], where it is also shown that IPS is equivalent to the so-called simple product ips in which selection functions $\varepsilon_s = \varepsilon_{|s|}$ can only depend on the *length* of s. Moreover (as we recalled in Chapter 9) Berger [10] shows that BBC is definable from UR, which is in turn an instance of OR. The difficulty that lies ahead is in relating these two groups of functionals, namely those based on bar recursion proper (IPS, MBR) with those based on open recursion (BBC, UR, OR). We manage to do this in both directions, proving that $\mathsf{OR}_<$ is definable from IPS (whenever the transfinite recursor $\mathsf{R}_<$ can be defined in system T), and conversely that the 'weakest' variant of open recursion, the BBC functional, defines ips. Combined with the results just stated, these complete the chain of equivalences.

Unlike previous chapters, we seem to require $\Delta :\equiv \mathsf{Cont} + \mathsf{QF\text{-}BI}$ in order to verify our constructions (this has also been the case for previous results involving the equivalence of implicit forms of bar recursion [12, 28]), so our equivalence results only holds in models that validate Δ. Of course Δ is admitted in the standard type structure $\mathscr{C}^\omega$ of total continuous functionals, so we believe that our dependence on Δ is reasonable. We leave open whether it is indeed possible to modify our definability results so that they are valid in type structures that do not satisfy Cont such as the strongly majorizable functionals.

We make no claims that our constructions are optimal at the level of types. For instance, it may well be that either in special cases, or even in general, our definition of OR from IPS can be done using a weaker instance of IPS than the one we have used. Optimality in this sense is a secondary concern here, and is of course not necessary at all to establish primitive recursive definability. However we believe that beneath the abstruse syntax that is inevitable for these kinds of result, our constructions are based on intuitive ideas. We define OR from IPS using a minimal bad sequence style algorithm (similar to, but simpler than the instance of EPS given in Chapter 8), and in turn ips is defined from BBC using a natural diagonal computation over infinite arrays of partial functions.

There are several factors that have motivated the research presented in this chapter. Each of the forms of recursion considered below have been used to give computational interpretations to variants of the axiom of choice, and so by comparing these forms of recursion we are able to give some insight into the differences between programs extracted using these forms of recursion. However, our main goal has been to continue the work carried out by various authors in the last few years ([11, 12, 30]) on establishing and classifying the known variants of recursion according to whether or not they are interdefinable. Our results are the first to relate bar recursive functionals with open recursive functionals in this sense.

The organisation of this chapter is rather straightforward. We begin in Section 12.1 by proving that IPS defines open recursion, and then outline the special case that IPS defines BBC in Section 12.2. Finally, we prove that BBC defines ips in Section 12.3.

12.1 Defining open recursion from IPS

We show that we can primitive recursively construct the open recursor $\mathsf{OR}_{\rho,\tau,<}$ from IPS for any decidable, well-founded relation on arbitrary type ρ, provided that we have access to the transfinite recursor $\mathsf{R}_{(\rho,<),\sigma}$ over $<$ of arbitrary type σ, which we recall is defined as

$$\mathsf{R}^g_{(\rho,<),\sigma}(x) \overset{\sigma}{:=} g_x(\lambda y\ .\ \mathsf{R}^g_<(y) \text{ if } y < x)$$

(note that here, as always, ‘ if $y < x$’ is shorthand for if $y < x$, else 0_σ). Whenever $\mathsf{R}_<$ is definable in T, it follows that $\mathsf{OR}_{\rho,\tau,<}$ is primitive recursively definable from IPS. We motivate our construction by sketching it in the simple case of open recursion on the Cantor space.

Illustration: Open recursion on the Cantor space $\mathbb{B}^{\mathbb{N}}$

The following discussion will be deliberately informal. In order to define open recursion $\mathsf{OR}_{\mathbb{B},\mathbb{N},<}$ on the Cantor space (that is sequences of booleans, lexicographically ordered on the relation $0 < 1$), we want to construct a term $\Phi := t(\mathsf{IPS})$ of type $(\mathbb{B}^{\mathbb{N}} \times (\mathbb{N} \times \mathbb{B}^{\mathbb{N}} \to \mathbb{N}) \to \mathbb{N}) \to (\mathbb{B}^{\mathbb{N}} \to \mathbb{N})$ which satisfies the following equation:

$$\Phi^F(u) \overset{\mathbb{N}}{:=} F_u(\lambda n, w\ .\ \Phi^F([u](n) * 0 @ w) \text{ if } u(n) = 1).$$

Our basic idea is to use an instance of IPS^u of type $\sigma := \mathbb{B}^{\mathbb{N}} \times (\mathbb{B}^{\mathbb{N}} \to \mathbb{N})$ (for some ε, q parametrised by u, F) to construct a sequence $\mathsf{IPS}^u(\langle\rangle) = \langle vs^u_0, fs^u_0\rangle, \langle vs^u_1, fs^u_1\rangle, \ldots$ where $\langle vs^u_n, fs^u_n\rangle : \sigma$.

(a) Each vs^u_n will represent a copy of u,

(b) Whenever $u(n) = 1$ the function fs^u_n will represent the function $\lambda w \,.\, \Phi^F([u](n) * 0 \mathbin{@} w)$ (and will just be the zero functional otherwise).

We construct parameters for IPS^u so that vs^u and fs^u satisfy (a) and (b). First, define $q^F \colon \sigma^{\mathbb{N}} \to \mathbb{N}$ by

$$q^F(vs^{(\mathbb{B}^{\mathbb{N}})^{\mathbb{N}}}, fs^{(\mathbb{B}^{\mathbb{N}} \to \mathbb{N})^{\mathbb{N}}}) := F_{\widetilde{vs}}(\lambda n, w \,.\, fs_n w).$$

where $\widetilde{vs}$ is the diagonal sequence $\lambda n \,.\, vs^u_n(n)$. Secondly, given $s \colon \mathbb{B}^*$ and $v \colon \mathbb{B}^{\mathbb{N}}$, define the selection function $\varepsilon_{s,v} \colon (\sigma \to \mathbb{N}) \to \sigma$ by

$$\varepsilon_{s,v}(p^{\sigma \to \mathbb{N}}) \overset{\mathbb{B}^{\mathbb{N}} \times (\mathbb{B}^{\mathbb{N}} \to \mathbb{N})}{:=} \begin{cases} \langle v, \lambda w \,.\, p(s * 0 \mathbin{@} w, 0_{\mathbb{B}^{\mathbb{N}} \to \mathbb{N}}) \rangle & \text{if } v(|s|) = 1 \\ \langle v, 0 \rangle & \text{otherwise,} \end{cases}$$

and from this define the dependent selection function $\tilde{\varepsilon}^u \colon \sigma^* \to J_{\mathbb{N}}\sigma$ by

$$\tilde{\varepsilon}^u_{\langle\rangle}(p) := \varepsilon_{\langle\rangle, u}(p)$$

$$\tilde{\varepsilon}^u_{\langle v_0, f_0 \rangle, \dots, \langle v_{n-1}, f_{n-1} \rangle}(p) := \varepsilon_{\langle v_0(0), \dots, v_{n-1}(n-1) \rangle, v_{n-1}}(p).$$

Then we claim that we can define $\Phi(u) := q^F(\mathsf{IPS}^{\tilde{\varepsilon}^u, q^F}(\langle\rangle))$. Let $\langle vs^u, fs^u \rangle = \mathsf{IPS}^{\tilde{\varepsilon}^u, q^F}(\langle\rangle)$. By the characterising equations for IPS (9.3) $vs^u_0 = u$ and $vs^u_n = vs^u_{n-1}$ for all $n > 0$, therefore by induction we must have $vs^u_n = u$ for all n and in particular $\widetilde{vs^u} = u$, which means that

$$\Phi(u) = q^F(vs^u, fs^u) = F_u(\lambda n, w \,.\, fs_n w)$$

Now we want to show that whenever $u(n) = 1$, we have $fs_n w = \Phi([u](n) * 0 \mathbin{@} w)$. By the second characterising equation (9.3), and the fact that $u(n) = 1 \to vs_{n-1}(n) = 1$ and $\langle vs_0(0), \dots, vs_{n-1}(n-1) \rangle = [u](n)$ we have

$$fs_n w = q^F(\mathsf{IPS}^{\tilde{\varepsilon}^u, q^F}([vs^u, fs^u](n) * \langle [u](n) * 0 \mathbin{@} w, 0 \rangle)).$$

Our main claim is that (using the abbreviation $t = [u](n) * 0 \mathbin{@} w$)

$$(\dagger) \quad q^F(\mathsf{IPS}^{\tilde{\varepsilon}^u, q^F}([vs^u, fs^u](n) * \langle t, 0 \rangle)) = q^F(\mathsf{IPS}^{\tilde{\varepsilon}^t, q^F}([vs^t, fs^t](n) * \langle t, 0 \rangle))$$

where $\langle vs^t, fs^t \rangle \overset{\sigma^{\mathbb{N}}}{:=} \mathsf{IPS}^{\tilde{\varepsilon}^t, q^F}(\langle\rangle)$. If we can prove ($\dagger$), then the result follows, since by the first characteristic equation and the fact that $vs^t_{n-1}(n) = t(n) = 0$ we clearly have $\langle vs^t_n, fs^t_n \rangle = \langle t, 0 \rangle$, so that (by Lemma 9.8)

$$q^F(\mathsf{IPS}^{\tilde{\varepsilon}^t, q^F}([vs^t, fs^t](n) * \langle t, 0 \rangle)) = q^F(\mathsf{IPS}^{\tilde{\varepsilon}^t, q^F}([vs^t, fs^t](n+1))) = q^F(\mathsf{IPS}^{\tilde{\varepsilon}^t, q^F}(\langle\rangle))$$

and therefore $fs_n w = \Phi(t) = \Phi([u](n) * 0 \mathbin{@} w)$. The claim ($\dagger$) is by no means obvious, but it is intuitively true for the following reasons. Firstly, if we assume inductively that

$$fs^u_m w' = \Phi([u](m) * 0 \mathbin{@} w') \text{ if } u(m) = 1 \qquad fs^t_m w' = \Phi([t](m) * 0 \mathbin{@} w') \text{ if } t(m) = 1$$

for $m < n$ and arbitrary w' then we can equate fs^u_m and fs^t_m since $[u](m+1) = [t](m+1)$, and thereby reduce (†) to

$$(\dagger\dagger) \quad q^F(\mathsf{IPS}^{\tilde{\varepsilon}^u, q^F}([vs^u, \mathit{fs}^u](n) * \langle t, 0\rangle)) = q^F(\mathsf{IPS}^{\tilde{\varepsilon}^t, q^F}([vs^t, \mathit{fs}^u](n) * \langle t, 0\rangle))$$

Now the two sides of the (††) are evaluated by computing optimal extensions to the sequences $[vs^u, \mathit{fs}^u](n) * \langle t, 0\rangle$ and $[vs^t, \mathit{fs}^u](n) * \langle t, 0\rangle$ respectively. The crucial point is that to do this q^F, $\tilde{\varepsilon}^t$ and $\tilde{\varepsilon}^u$ only ever look at the first m values of vs^u_m and vs^t_m respectively, and $[vs^u_m](m) = [vs^t_m](m)$ for all $m < n$. Therefore the two sides must be equal.

We note that this reasoning only works in the continuous type-structures. In non-continuous structures the two sides of (††) could well be interpreted as different elements, despite that fact that they satisfy the same recursive equations. We require Cont to ensure that they evaluate to the same thing.

Now we give the full definability result. We generalise our construction so that it works for arbitrary types and well-founded relations, and give a detailed proof of the correctness of our term (unlike the rather sketchy illustration above). Unsurprisingly, the proof of our claim (†) involves a fairly lengthy induction argument, which we separate from the main proof as Lemma 12.2.

The general construction

We define the type $\sigma_{\rho,\tau} := \rho^{\mathbb{N}} \times (\rho \times \rho^{\mathbb{N}} \to \tau)$.

Theorem 12.1. *Let ρ be an arbitrary type, and τ a discrete type. Then $\mathsf{OR}_{(\rho,<),\tau}$ is $\mathsf{T} + \mathsf{R}_{<}$ definable from $\mathsf{IPS}_{\sigma_{\rho,\tau},\tau}$, provably in $\mathsf{Cont} + \mathsf{QF\text{-}BI}$.*

Proof. Suppose that $F \colon \rho^{\mathbb{N}} \times (\mathbb{N} \times \rho \times \rho^{\mathbb{N}} \to \tau) \to \tau$ is a parameter for open recursion. We define $q^F : \sigma^{\mathbb{N}} \to \tau$ by

$$q^F(vs, \mathit{fs}) \overset{\tau}{:=} F_{\widetilde{vs}}(\lambda n, y, v \, . \, \mathit{fs}_n y v).$$

for $vs \colon (\rho^{\mathbb{N}})^{\mathbb{N}}$ and $\mathit{fs} \colon (\rho \times \rho^{\mathbb{N}} \to \tau)^{\mathbb{N}}$, where $\widetilde{vs} \overset{\rho^{\mathbb{N}}}{:=} \lambda n \, . \, vs_n(n)$. For the selection functions, first define $\varepsilon_{s,v} \colon J_\tau \sigma$ for $s \colon \rho^*$, $v \colon \rho^{\mathbb{N}}$ by

$$\varepsilon_{s,v}(p^{\sigma \to \tau}) \overset{\sigma}{:=} \langle v, f^{p,s}_{v(|s|)} \rangle$$

where $f^{p,s}$ is defined as in Chapter 8 using $\mathsf{R}_{(\rho,<),\rho \times \rho^{\mathbb{N}} \to \tau}$ as

$$f^{p,s}_x \overset{\rho \times \rho^{\mathbb{N}} \to \tau}{:=} \lambda y, v \, . \, p(s * y \mathbin{@} v, f^{p,s}_y) \text{ if } y < x.$$

Note that if x is minimal with respect to $<$ then $f^{p,s}_x = 0$. Then we can define, for $u \colon \rho^{\mathbb{N}}$, $\tilde{\varepsilon}^u \colon \sigma^* \to J_\tau \sigma$ by

$$\tilde{\varepsilon}^u_{\langle\rangle}(p) := \varepsilon_{\langle\rangle,u}(p)$$

$$\tilde{\varepsilon}^u_{\langle\langle v_0, f_0\rangle \ldots, \langle f_{n-1}, v_{n-1}\rangle\rangle}(p) := \varepsilon_{\langle v_0(0), \ldots, v_{n-1}(n-1)\rangle, v_{n-1}}(p).$$

Finally, let us define

$$\Phi^F(u) := q^F(\mathsf{IPS}^{\tilde{\varepsilon}^u, q^F}_{\sigma,\tau}(\langle\rangle)).$$

We claim that $\mathsf{OR}_{(\rho,<),\tau}[\Phi]$. Let us use the abbreviation $\langle vs^u, fs^u\rangle \overset{\sigma^{\mathbb{N}}}{=} \mathsf{IPS}^{\tilde{\varepsilon}^u, q^F}(\langle\rangle)$ where $vs^u \colon (\rho^{\mathbb{N}})^{\mathbb{N}}$ and $fs^u \colon (\rho \times \rho^{\mathbb{N}} \to \tau)^{\mathbb{N}}$. First we prove that

(a) for arbitrary u, we have $vs^u_n = u$ and so in particular $\widetilde{vs^u} = u$.

This is easily verified by induction and the characterising equations (9.3). First, let us use the notation

$$p_{n,u} = p_{[vs^u, fs^u](n)} := \lambda\langle w, g\rangle^\sigma \,.\, q^F(\mathsf{IPS}^{\tilde{\varepsilon}^u, q^F}([vs^u, fs^u](n) * \langle w, g\rangle)).$$

Then by (9.3) $vs^u_0 = \tilde{\varepsilon}^u_{\langle\rangle}(p_{0,u})_0 = u$, and $vs^u_n = \tilde{\varepsilon}^u_{[vs^u, fs^u](n)}(p_{n,u})_0 = vs^u_{n-1}$ for $n > 0$, so we have $vs^u_n = u$ for all n. Secondly, we wish to prove

(b) $fs^u_n yv = q^F(\mathsf{IPS}^{\tilde{\varepsilon}^{[u](n)*y \,@\, v}, q^F}(\langle\rangle))$ if $y < u(n)$.

The result then follows, because

$$\begin{aligned}
\Phi^F(u) &= q^F(vs^u, fs^u) \\
&= F_{\widetilde{vs^u}}(\lambda n, y, v \,.\, fs^u_n yv) \\
&\overset{(a),(b)}{=} F_u(\lambda n, y, v \,.\, q^F(\mathsf{IPS}^{\tilde{\varepsilon}^{[u](n)*y \,@\, v}, q^F}(\langle\rangle)) \text{ if } y < u(n)) \\
&= F_u(\lambda n, y, v \,.\, \Phi([u](n) * y \,@\, v) \text{ if } y < u(n)).
\end{aligned}$$

The main difficulty of the proof lies in establishing (b), for which we need the following additional claim:

(c) whenever $[u](n) = [t](n)$ for arbitrary u, t we have $p_{n,u} = p_{n,t}$.

This is proved as Lemma 12.2 below. Continuing for now, by (9.3) and (a) again we have

$$fs^u_0 = \tilde{\varepsilon}^u_{\langle\rangle}(p_{0,u})_1 = f^{p_{0,u},\langle\rangle}_{u(0)} \quad fs^u_n = \tilde{\varepsilon}^u_{[vs^u, fs^u](n)}(p_{n,u})_1 = \tilde{\varepsilon}^u_{[u](n),u}(p_{n,u}) = f^{p_{n,u},[u](n)}_{u(n)}$$

therefore for arbitrary arguments y, v we have

$$\begin{aligned}
fs^u_n yv &= f^{p_{n,u},[u](n)}_{u(n)} yv \\
&= p_{n,u}([u](n) * y \,@\, v, f^{p_{n,u},[u](n)}_y) \text{ if } y < u(n) \\
&\overset{Lem.12.2}{=} p_{n,[u](n)*y \,@\, v}([u](n) * y \,@\, v, f^{p_{n,[u](n)*y \,@\, v},[u](n)}_y) \text{ if } y < u(n) \\
&= p_{n,t}(t, f^{p_{n,t},[t](n)}_{t(n)}) \text{ if } t(n) < u(n)
\end{aligned}$$

where for the last equation we substitute the abbreviation $t := [u](n) * y \,@\, v$. If we had $n = 0$, then

$$p_{0,t}(t, f^{p_{0,t},[t](0)}_{t(0)}) = p_{0,t}(\tilde{\varepsilon}^t_{\langle\rangle}(p_{0,t})) \overset{(9.3)}{=} q^F(vs^t, fs^t)$$

and otherwise

$$\begin{aligned} p_{n,t}(t, f_{t(n)}^{p_{n,t},[t](n)}) &= p_{n,t}(\varepsilon_{[t](n),t}(p_{n,t})) \\ &\overset{(a)}{=} p_{n,t}(\tilde{\varepsilon}^{t}_{[vs^t,fs^t](n)}(p_{n,t})) \\ &\overset{(9.3)}{=} q^F(vs^t, fs^t). \end{aligned}$$

Therefore we have established $p_{n,t}(t, f_{t(n)}^{p_{n,t},[t](n)}) = q^F(vs^t, fs^t) = q^F(vs^{[u](n)*y \text{ @ } v}, fs^{[u](n)*y \text{ @ } v})$ for all n, y and v, and therefore

$$fs^u_n yv = q^F(\mathsf{IPS}^{\tilde{\varepsilon}^{[u](n)*y \text{ @ } v}, q^F}(\langle\rangle)) \text{ if } y < u(n).$$

□

It remains to prove (c), and then we're done. This involves a lengthy, but fairly routine proof by combined induction and bar induction. It is here that we seem to require continuity.

Lemma 12.2. *For arbitrary sequences u, t of type $\rho^{\mathbb{N}}$, whenever $[u](n) = [t](n)$ we have $p_{n,u} = p_{n,t}$, provably in* $\mathsf{Cont} + \mathsf{QF\text{-}BI}$.

Proof. The proof is by induction on the formula

$$A(n) :\equiv \forall u, t([u](n) = [t](n) \to p_{n,u} = p_{n,t}).$$

To establish $A(0)$ we must show that $p_{0,u} = p_{0,t}$ for arbitrary u, t. Given arguments g, w, we run bar induction of type σ on the quantifier-free formula

$$P^0(\langle ws, gs\rangle^{\sigma^*}) :\equiv (q^F(\mathsf{IPS}^{\tilde{\varepsilon}^u, q^F}(\langle w, g\rangle * \langle ws, gs\rangle)) \overset{\tau}{=} q^F(\mathsf{IPS}^{\tilde{\varepsilon}^t, q^F}(\langle w, g\rangle * \langle ws, gs\rangle))),$$

then $P(\langle\rangle)$ yields $p_{0,u}wg = p_{0,t}wg$. Given $\langle vs, fs\rangle \colon \sigma^{\mathbb{N}}$, by Cont applied to $q_{\langle w,g\rangle} := \lambda\langle vs, fs\rangle \,.\, q(\langle w, g\rangle * \langle vs, fs\rangle)$ there exists N such that

$$\begin{aligned} q(\mathsf{IPS}^{\varepsilon^u}(\langle w, g\rangle * [vs, fs](N))) &= q(\langle w, g\rangle * [vs, fs](N) \text{ @ } \mathsf{IPS}^{\tilde{\varepsilon}^u}(\langle w, g\rangle * [vs, fs](N))) \\ &\overset{\mathsf{Cont}}{=} q(\langle w, g\rangle * [vs, fs](N) \text{ @ } \mathsf{IPS}^{\tilde{\varepsilon}^t}(\langle w, g\rangle * [vs, fs](N))) \\ &= q(\mathsf{IPS}^{\varepsilon^t}(\langle w, g\rangle * [vs, fs](N))) \end{aligned}$$

and for the bar induction step we have

$$\begin{aligned} q^F(\mathsf{IPS}^{\varepsilon^u}(\langle w, g\rangle * \langle ws, gs\rangle)) &= q^F(\mathsf{IPS}^{\tilde{\varepsilon}^u}(\langle w, g\rangle * \langle ws, gs\rangle * a_u)) \\ &\overset{(*)}{=} q^F(\mathsf{IPS}^{\tilde{\varepsilon}^u}(\langle w, g\rangle * \langle ws, gs\rangle * a_t)) \\ &\overset{B.I.H.}{=} q^F(\mathsf{IPS}^{\tilde{\varepsilon}^t}(\langle w, g\rangle * \langle ws, gs\rangle * a_t)) \\ &\overset{L.9.8}{=} q^F(\mathsf{IPS}^{\tilde{\varepsilon}^t}(\langle w, g\rangle * \langle ws, gs\rangle)) \end{aligned}$$

where for $(*)$ we have

$$\begin{aligned}
a_u &= \tilde{\varepsilon}^u_{\langle w,g\rangle * \langle ws,gs\rangle}(\lambda b^\sigma \,.\, q(\mathsf{IPS}^{\tilde{\varepsilon}^u}(\langle w,g\rangle * \langle ws,gs\rangle * b)))\\
&\overset{B.I.H.}{=} \varepsilon_{\langle w(0),ws_0(1),\dots,ws_{m-1}(m)\rangle, ws_{m-1}}(\lambda b^\sigma \,.\, q(\mathsf{IPS}^{\tilde{\varepsilon}^t}(\langle w,g\rangle * \langle ws,gs\rangle * b)))\\
&= \tilde{\varepsilon}^t_{\langle w,g\rangle * \langle ws,gs\rangle}(\lambda b^\sigma \,.\, q(\mathsf{IPS}^{\tilde{\varepsilon}^t}(\langle w,g\rangle * \langle ws,gs\rangle * b)))\\
&= a_t,
\end{aligned}$$

where $m = |ws|$ and for $m = 0$ we replace ws_{m-1} by w. Therefore, by bar induction we have $A(0)$. The main induction step follows a similar pattern. For $n > 0$ we derive $A(n)$, assuming that $A(m)$ holds for all $m < n$.

Suppose that $[u](n) = [t](n)$. For arbitrary w, g we run bar induction on the formula

$$\begin{cases} P^n(\langle ws,gs\rangle) :\equiv \\ (q^F(\mathsf{IPS}^{\tilde{\varepsilon}^u,q^F}([vs^u,fs^u](n) * \langle w,g\rangle * \langle ws,gs\rangle) \overset{\tau}{=} q^F(\mathsf{IPS}^{\tilde{\varepsilon}^t,q^F}([vs^t,fs^t](n) * \langle w,g\rangle * \langle ws,gs\rangle)), \end{cases}$$

then $P^n(\langle\rangle)$ yields $p_{n,u}wg = p_{n,t}wg$. The main induction hypothesis is used to verify that $fs^u_m = fs^t_m$ for all $m < n$. To see this, note that for $0 < m < n$

$$\begin{aligned}
fs^u_m &= \tilde{\varepsilon}^u_{[vs^u,fs^u](m)}(p_{m,u})_1\\
&= \varepsilon_{\langle vs^u_0(0),\dots,vs^u_{m-1}(m-1)\rangle, vs^u_{m-1}}(p_{m,u})_1\\
&\overset{(a)}{=} f^{p_{m,u},[u](m)}_{u(m)}\\
&\overset{I.H.}{=} f^{p_{m,t},[t](m)}_{t(m)}\\
&\overset{(a)}{=} \varepsilon_{\langle vs^t_0(0),\dots,vs^t_{m-1}(m-1)\rangle, vs^t_{m-1}}(p_{m,t})_1\\
&= \tilde{\varepsilon}^t_{[vs^t,fs^t](m)}(p_{m,t})_1\\
&= fs^t_m.
\end{aligned}$$

Similarly for $m = 0$ we get $fs^u_0 = \varepsilon_{\langle\rangle,u}(p_{0,u})_1 = f^{p_{0,u},\langle\rangle}_{u(0)} \overset{I.H.}{=} f^{p_{0,t},\langle\rangle}_{t(0)} = fs^t_0$. Now we continue as before. Given $\langle vs,fs\rangle \colon \sigma$, by Cont applied to $q_{[vs^u,fs^u](n)*\langle w,g\rangle}$ there exists N such that

$$\begin{aligned}
&q^F(\mathsf{IPS}^{\tilde{\varepsilon}^u,q^F}([vs^u,fs^u](n) * \langle w,g\rangle * [vs,fs](N)))\\
&= q^F([vs^u,fs^u](n) * \langle w,g\rangle * [vs,fs](N) \mathbin{@} \mathsf{IPS}^{\tilde{\varepsilon}^u,q^F}([vs^u,fs^u](n) * \langle w,g\rangle * [vs,fs](N)))\\
&\overset{\mathsf{Cont}}{=} q^F([vs^u,fs^u](n) * \langle w,g\rangle * [vs,fs](N) \mathbin{@} \mathsf{IPS}^{\tilde{\varepsilon}^t,q^F}([vs^t,fs^t](n) * \langle w,g\rangle * [vs,fs](N)))\\
&= F_{[vs^u](n)*\widetilde{w*[vs]}(N) \mathbin{@} vs'}([fs^u](n) * g * [fs](N) \mathbin{@} fs')\\
&\overset{(\dagger)}{=} F_{[vs^t](n)*\widetilde{w*[vs]}(N) \mathbin{@} vs'}([fs^t](n) * g * [fs](N) \mathbin{@} fs')\\
&= q^F([vs^t,fs^t](n) * \langle w,g\rangle * [vs,fs](N) \mathbin{@} \mathsf{IPS}^{\tilde{\varepsilon}^t,q^F}([vs^t,fs^t](n) * \langle w,g\rangle * [vs,fs](N)))\\
&= q^F(\mathsf{IPS}^{\tilde{\varepsilon}^t,q^F}([vs^t,fs^t](n) * \langle w,g\rangle * [vs,fs](N))),
\end{aligned}$$

where we abbreviate $\langle vs', fs'\rangle = \mathsf{IPS}^{\tilde{\varepsilon}^t, q^F}([vs^t, fs^t](n) * \langle w, g\rangle * [vs, fs](N))$, and (†) holds because $[fs^u](n) = [fs^t](n)$ and because for $m < n$ we have $vs^u_m(m) = u(m) = t(m) = vs^t_m(m)$ (by (a) as in Theorem 12.1) and therefore $\widetilde{[vs^u](n)} = \widetilde{[vs^t](n)}$. So we have established that for all $\langle vs, fs\rangle$ there exists N such that $P^n([vs, fs](N))$ holds.

The bar induction step is routine:

$$\begin{aligned}
& q^F(\mathsf{IPS}^{\tilde{\varepsilon}^u}([vs^u, fs^u](n) * \langle w, g\rangle * \langle ws, gs\rangle)) \\
&= q^F(\mathsf{IPS}^{\tilde{\varepsilon}^u}([vs^u, fs^u](n) * \langle w, g\rangle * \langle ws, gs\rangle * a_u)) \\
&\overset{(*)}{=} q^F(\mathsf{IPS}^{\tilde{\varepsilon}^u}([vs^u, fs^u](n) * \langle w, g\rangle * \langle ws, gs\rangle * a_t)) \\
&\overset{B.I.H.}{=} q^F(\mathsf{IPS}^{\tilde{\varepsilon}^t}([vs^t, fs^t](n) * \langle w, g\rangle * \langle ws, gs\rangle * a_t)) \\
&= q^F(\mathsf{IPS}^{\tilde{\varepsilon}^t}([vs^t, fs^t](n) * \langle w, g\rangle * \langle ws, gs\rangle))
\end{aligned}$$

where for $(*)$ we have

$$\begin{aligned}
a_u &= \tilde{\varepsilon}^u_{[vs^u, fs^u](n) * \langle w, g\rangle * \langle ws, gs\rangle}(\lambda b \,.\, q^F(\mathsf{IPS}^{\tilde{\varepsilon}^u}([vs^u, fs^u](n) * \langle w, g\rangle * \langle ws, gs\rangle * b)) \\
&= \varepsilon_{\widetilde{[vs^u](n)} * w * ws, ws_{m-1}}(\lambda b \,.\, q^F(\mathsf{IPS}^{\tilde{\varepsilon}^u}([vs^u, fs^u](n) * \langle w, g\rangle * \langle ws, gs\rangle * b)) \\
&\overset{B.I.H.}{=} \varepsilon_{\widetilde{[vs^u](n)} * w * ws, ws_{m-1}}(\lambda b \,.\, q^F(\mathsf{IPS}^{\tilde{\varepsilon}^t}([vs^t, fs^t](n) * \langle w, g\rangle * \langle ws, gs\rangle * b)) \\
&\overset{(**)}{=} \varepsilon_{\widetilde{[vs^t](n)} * w * ws, ws_{m-1}}(\lambda b \,.\, q^F(\mathsf{IPS}^{\tilde{\varepsilon}^t}([vs^t, fs^t](n) * \langle w, g\rangle * \langle ws, gs\rangle * b)) \\
&= \tilde{\varepsilon}^t_{[vs^t, fs^t](n) * \langle w, g\rangle * \langle ws, gs\rangle}(\lambda b \,.\, q^F(\mathsf{IPS}^{\tilde{\varepsilon}^t}([vs^t, fs^t](n) * \langle w, g\rangle * \langle ws, gs\rangle * b)) \\
&= a_t
\end{aligned}$$

where of course $(**)$ follows because $\widetilde{[vs^u](n)} = \widetilde{[vs^t](n)}$, and as before $m = |ws|$, with ws_{m-1} replaced by w for $m = 0$. This completes the bar induction, and therefore we obtain $\forall m < nA(m) \to A(n)$, and so by induction we have $\forall nA(n)$, which completes the proof. □

12.2 Defining UI and BBC from IPS

An immediate consequence of the previous section is the following.

Theorem 12.3. *Both update recursion and the* BBC *functional are primitive recursively definable from* IPS*, over* $\mathsf{Cont} + \mathsf{QF\text{-}BI}$.

Proof. By Proposition 9.20, $\mathsf{UR}_{\rho,\tau}$ is definable from an instance of $\mathsf{OR}_{(\bar{\rho},<),\tau}$ on the relation $(a, x^\rho) < (b, y) :\equiv (a = 1 \wedge b = 0)$. Clearly in this case $\mathsf{R}_<$ is trivially definable in T, therefore by Theorem 12.1 $\mathsf{UR}_{\rho,\tau}$ is primitive recursively definable from an instance of $\mathsf{IPS}_{\sigma_{\bar{\rho},\tau},\tau}$. By Proposition 9.16, an instance of IPS of the same type defines $\mathsf{BBC}_{\rho,\tau}$. □

Moreover by combining the proofs of these definability results we obtain an explicit construction of UR and BBC from IPS. We state the construction of the latter below. Given parameters $q\colon \rho^{\mathbb{N}} \to \tau$ and $\varepsilon\colon \mathbb{N} \to J_\tau\rho$ for BBC, for $\sigma := \bar{\rho}^{\mathbb{N}} \times (\bar{\rho} \times \bar{\rho}^{\mathbb{N}} \to \tau)$, define $q\colon \sigma^{\mathbb{N}} \to \tau$ and $\tilde{\varepsilon}\colon \bar{\rho}^{\mathbb{N}} \to (\sigma^* \to J_\tau\sigma)$ by

$$\tilde{q}(vs, fs) \overset{\tau}{:=} q(\widetilde{vs} \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, fs_n(1,x)\tilde{vs}))$$

$$\tilde{\varepsilon}^u_{\langle\rangle}(p) \overset{\sigma}{:=} \begin{cases} \langle u, f^{p,\langle\rangle}\rangle & \text{if } 0 \notin \mathrm{dom}(u) \\ \langle u, 0\rangle & \text{otherwise.} \end{cases}$$

$$\tilde{\varepsilon}^u_{\langle v_0,f_0\rangle,\dots,\langle v_{n-1},f_{n-1}\rangle}(p) \overset{\sigma}{:=} \begin{cases} \langle v_{n-1}, f^{p,\langle v_0(0),\dots,v_{n-1}(n-1)\rangle}\rangle & \text{if } n \notin \mathrm{dom}(v_{n-1}) \\ \langle v_{n-1}, 0\rangle & \text{otherwise.} \end{cases}$$

where $f^{p,s} := \lambda y, v \,.\, p(s*(1,y) \mathbin{@} v, 0)$. Then $\Phi^{\varepsilon,q}(u) := \tilde{q}(\mathsf{IPS}^{\tilde{\varepsilon}^u,\tilde{q}}(\langle\rangle))$ satisfies $\mathsf{sBBC}_{\rho,\tau}[\Phi]$. Following the proof of Theorem 12.1 we see that

$$\begin{aligned}
\Phi^{\varepsilon,q}(u) &= \tilde{q}(vs^u, fs^u) \\
&= q(\widetilde{vs^u} \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, fs^u_n(1,x)\widetilde{vs^u})) \\
&\overset{(a)}{=} q(u \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, fs^u_n(1,x)u)) \\
&\overset{(b)}{=} q(u \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, \Phi([u](n) * (1,x) \mathbin{@} u))) \\
&= q(u \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, \Phi(u^x_n)))
\end{aligned}$$

and (b) follows because whenever $n \notin \mathrm{dom}(u)$

$$\begin{aligned}
fs^u_n(1,x)u &= f^{p_{n,u},[u](n)}(1,x)u \\
&= p_{n,u}([u](n) * (1,x) \mathbin{@} u, 0) \\
&\overset{L.12.2}{=} p_{n,u^x_n}(u^x_n, 0) \\
&= p_{n,u^x_n}(\tilde{\varepsilon}^{u^x_n}_{[vs^{u^x_n}, fs^{u^x_n}](n)}(p_{n,u^x_n})) \\
&= q(vs^{u^x_n}, fs^{u^x_n}) \\
&= \Phi(u^x_n).
\end{aligned}$$

12.3 Defining IPS from BBC

Having proved that IPS defines BBC, we now establish the converse. The problem we face is to capture the dependency implicit in IPS with BBC, which constructs a sequence symmetrically, each element computed independently of the others. Due to a result of Escardó and Oliva we can at least eliminate the dependency in the selection functions of IPS, by reducing IPS to the simple product of selection functions ips. Given a family of simple selection functions $\varepsilon\colon \mathbb{N} \to J_\tau\rho$, define the functional $\mathsf{ips}_{\rho,\tau}$ by

$$\mathsf{ips}^{\varepsilon,q}(s) := s \mathbin{@} \mathsf{ips}^{\varepsilon,q}(s * a_s)$$

where $a_s := \varepsilon_{|s|}(\lambda x \,.\, q(\mathsf{ips}^{\varepsilon,q}(s * x)))$. Note that as with IPS, ips can be equivalently defined as

$$\mathsf{ips}^{\varepsilon,q}(s) := s \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, q(\mathsf{ips}^{\varepsilon,q}(t_n * x)))$$

where $t_n := [\mathsf{ips}(s)](n)$. It turns out that the simple product of selection functions ips actually defines the general product IPS.

Proposition 12.4 ([28])**.** *IPS and ips are primitive recursively equivalent over $\mathsf{Cont} + \mathsf{QF\text{-}BI}$.*

Note that technically this result is proved for variants sIPS, sips of IPS and ips. We confirm that our variants are equivalent to these in Appendix A.

Illustration: finite bar recursion

We now give a short and very informal illustration in the finite case, and show how we can simulate the finite product $\mathsf{fP}^{\varepsilon,q,3}_{\mathbb{N},\mathbb{N}}$ of length three with a finite variant of BBC. We abbreviate elements of $\bar{\mathbb{N}}$ as $(0,x) = \bot$ and $(1,x) = x$. Note that $\mathsf{fP}^{\varepsilon,q} = \langle X, Y[X], Z[X,Y]\rangle$ where

$$\begin{aligned} X &= \varepsilon_0(\lambda x \,.\, q(x, Y[x], Z[x, Y[x]]) \\ Y[x] &= \varepsilon_1(\lambda y \,.\, q(x, y, Z[x,y]) \\ Z[x,y] &= \varepsilon_2(\lambda z \,.\, q(x,y,z) \end{aligned}$$

We can similarly define a finite variant fBBC^3 of the BBC functional as

$$\mathsf{fBBC}^{\varepsilon,q,3}_{\rho,\mathbb{N}}(u^{\bar{\rho}^{(3)}}) \overset{\rho^{(3)}}{:=} u \mathbin{@} (\varepsilon_0(p_0), \varepsilon_1(p_1), \varepsilon_2(p_2))$$

where $p_i = \lambda x^\rho . q(\mathsf{fBBC}(u^x_i))$. We define $\mathsf{fP}^{\varepsilon,q}_{\mathbb{N},\mathbb{N}}$ with an instance $\mathsf{fBBC}^{\tilde{\varepsilon},\tilde{q}}_{\rho,\mathbb{N}}$ of fBBC type $\rho = \bar{\mathbb{N}}^{(3)}$, which outputs a 3×3 matrix (c_0, c_1, c_2) (where c_i denotes the ith column) whose entries lie in $\bar{\mathbb{N}}$, and takes as input a partial (and potentially empty) list of columns of this matrix. Note that in this instance of fBBC, there are two kinds of 'undefined': individual entries in a column can be undefined (and we write $\bot$), and also columns themselves can be undefined (which we write as $\bot_c$). We define parameters for fBBC as

$$\begin{aligned} \tilde{\varepsilon}_i(p) &\overset{\mathbb{N}^{(3)}}{=} \lambda n < 3 \,.\, \varepsilon_n(\lambda x \,.\, p(\langle [\tilde{\varepsilon}_i(p)](n) * x\rangle)) \\ \tilde{q}(c_0, c_1, c_2) &\overset{\mathbb{N}}{:=} q(d(c_0, c_1, c_2)) \\ d(c_0, c_1, c_2) &\overset{\mathbb{N}^{(3)}}{:=} \lambda n < 3 \,.\, \begin{cases} c_{n,m} & \text{for least } m \leq n \text{ with } n \neq \bot \\ 0 & \text{if no such } m \text{ exists.} \end{cases} \end{aligned}$$

Note that for the first equation to type-check, technically $\tilde{\varepsilon}_i(p)$ should be viewed as a total element of $\bar{\mathbb{N}}^{(3)}$, and also the finite sequence $[\tilde{\varepsilon}_i(p)](n) * x$ is extended using $\bot$ to a column $\bar{\mathbb{N}}^{(3)}$ if necessary.

We claim that $\mathsf{fP}^{\varepsilon,q}(\langle\rangle) = d(\mathsf{fBBC}^{\tilde{\varepsilon},\tilde{q}}(\bot_c, \bot_c, \bot_c))$. Writing $\mathsf{fBBC}(\bot_c, \bot_c, \bot_c) = (c_0, c_1, c_2)$ we get $d(c_0, c_1, c_2) = c_0$ (since $c_0 = \tilde{\varepsilon}_0(p_0)$ is total) where

$$\begin{aligned}
c_{0,0} &= \varepsilon_0(\lambda x \;.\; q(d(\mathsf{fBBC}(\langle x, \bot, \bot\rangle, \bot_c, \bot_c)))\\
c_{1,0} &= \varepsilon_1(\lambda y \;.\; q(d(\mathsf{fBBC}(\langle c_{0,0}, y, \bot\rangle, \bot_c, \bot_c)))\\
c_{2.0} &= \varepsilon_2(\lambda z \;.\; q(d(\mathsf{fBBC}(\langle c_{0,0}, c_{1,0}, z\rangle, \bot_c, \bot_c))).
\end{aligned}$$

Now, $c_{2,0} = \varepsilon_2(\lambda z \;.\; q(c_{0,0}, c_{1,0}, z)) = Z[c_{0,0}, c_{1,0}]$. To evaluate $c_{1,0}$ must evaluate

$$d(\mathsf{fBBC}(\langle c_{0,0}, y, \bot\rangle, \bot_c, \bot_c)) = d(\langle c_{0,0}, y, \bot\rangle \mathbin{@} (c'_0, c'_1, c'_2)) = \langle c_{0,0}, y, c'_{2,1}\rangle$$

where

$$\begin{aligned}
c'_{2,1} &= \varepsilon_2(\lambda z \;.\; q(d(\mathsf{fBBC}(\langle c_{0,0}, y, \bot\rangle, \langle c'_{0,1}, c'_{1,1}, z\rangle, \bot_c)))))\\
&= \varepsilon_2(\lambda z \;.\; q(\langle c_{0,0}, y, z\rangle))\\
&= Z[c_{0,0}, y].
\end{aligned}$$

Therefore we have $c_{1,0} = \varepsilon_1(\lambda y \;.\; q(\langle c_{0,0}, y, Z[c_{0,0}, y]\rangle)) = Y[c_{0,0}]$. Finally, we must evaluate

$$d(\mathsf{fBBC}(\langle x, \bot, \bot\rangle, \bot_c, \bot_c)) = d(\langle x, \bot, \bot\rangle \mathbin{@} (c''_0, c''_1, c''_2)) = \langle x, c''_{1,1}, c''_{2,1}\rangle.$$

Similar calculations give us

$$\begin{aligned}
c''_{2,1} &= \varepsilon_2(\lambda z \;.\; q(d(\mathsf{fBBC}(\langle x, \bot, \bot\rangle, \langle c''_{0,1}, c''_{1,1}, z\rangle, \bot_c)))\\
&= \varepsilon_2(\lambda z \;.\; q(\langle x, c''_{1,1}, z\rangle))))\\
&= Z[x, c''_{1,1}]\\
c''_{1,1} &= \varepsilon_1(\lambda y \;.\; q(d(\mathsf{fBBC}(\langle x, \bot, \bot\rangle, \langle c''_{0,1}, y, \bot\rangle, \bot_c))))\\
&\overset{(*)}{=} \varepsilon_1(\lambda y \;.\; q(\langle x, y, d_{2,2}\rangle)))\\
&= \varepsilon_1(\lambda y \;.\; q(\langle x, y, Z[x, y]\rangle))\\
&= Y[x]
\end{aligned}$$

where for $(*)$ we have $\mathsf{fBBC}(\langle x, \bot, \bot\rangle, \langle c''_{0,1}, y, \bot\rangle, \bot_c) = \langle x, \bot, \bot\rangle, \langle c''_{0,1}, y, \bot\rangle \mathbin{@} \langle d_0, d_1, d_2\rangle$. Backtracking, we obtain $c_{0,0} = \varepsilon_0(\lambda x \;.\; q(\langle x, Y[x], Z[x, y]\rangle)) = X$, and therefore $c_{1,0} = Y[X]$ and $c_{2,0} = Z[X, Y[X]]$. We illustrate the whole computation below.

$$\begin{aligned}
&\begin{pmatrix} x & & \\ \bot & & \\ \bot & & \end{pmatrix} \mapsto \begin{pmatrix} x & c''_{0,1} & \\ \bot & y & \\ \bot & \bot & \end{pmatrix} \mapsto \begin{pmatrix} x & c''_{0,1} & d_{0,2} \\ \bot & y & d_{1,2} \\ \bot & \bot & z \end{pmatrix} \overset{\varepsilon_2}{\mapsto} \begin{pmatrix} x & c''_{0,1} & d_{0,2} \\ \bot & y & d_{2,1} \\ \bot & \bot & Z[x,y] \end{pmatrix} \\
&\overset{\varepsilon_1}{\mapsto} \begin{pmatrix} x & c''_{0,1} & \\ \bot & Y[x] & \\ \bot & & \end{pmatrix} \\
&\mapsto \begin{pmatrix} x & c''_{0,1} & \\ \bot & Y[x] & \\ \bot & z & \end{pmatrix} \overset{\varepsilon_2}{\mapsto} \begin{pmatrix} x & c''_{0,1} & \\ \bot & Y[x] & \\ \bot & Z[x,Y[x]] & \end{pmatrix} \\
&\overset{\varepsilon_0}{\mapsto} \begin{pmatrix} X & & \\ & & \\ & & \end{pmatrix} \\
&\mapsto \begin{pmatrix} X & & \\ y & & \\ \bot & & \end{pmatrix} \mapsto \begin{pmatrix} X & c'_{0,1} & \\ y & c'_{1,1} & \\ \bot & z & \end{pmatrix} \overset{\varepsilon_2}{\mapsto} \begin{pmatrix} X & c'_{0,1} & \\ y & c'_{1,1} & \\ \bot & Z[X,y] & \end{pmatrix} \\
&\overset{\varepsilon_1}{\mapsto} \begin{pmatrix} X & & \\ Y[X] & & \\ & & \end{pmatrix} \\
&\mapsto \begin{pmatrix} X & & \\ Y[X] & & \\ z & & \end{pmatrix} \overset{\varepsilon_2}{\mapsto} \begin{pmatrix} X & & \\ Y[X] & & \\ Z[X,Y[X]] & & \end{pmatrix}
\end{aligned}$$

The idea behind the construction is that fBBC computes fP using a matrix. Whenever fP is 'testing' a variable in a column via a selection function, it carries the computation over to the column on the right. Once it has decided on this value, it continues the computation in the original column. In this sense it simulates a variant of fP_0^3 which comes equipped with markers indicating whether or not a variable is being checked, and the functional d can be viewed as an operation which removes those markers.

We can clearly extend our idea to a matrix of arbitrary size and type, and therefore simulate the general finite product fP_ρ^n using a finite form of BBC over an $n \times n$ matrix. The main result that follows is that our construction works just as well in an *unbounded* matrix, and that the unbounded product ips can be defined from an instance of BBC that computes an infinite array of partial functions.

The general construction

First we need some notation. Given a finite sequence $s\colon \rho^*$ let $\{s\}\colon \bar{\rho}^{\mathbb{N}}$ its standard embedding into the partial sequences i.e.

$$\{s\}(n) := \begin{cases} (1, s_n) & \text{if } n < |s| \\ (0, 0_\rho) & \text{otherwise.} \end{cases}$$

In the following we will often use a slight abuse of notation and just write s when it is obvious that we're treating it as an element of $\bar{\rho}^{\mathbb{N}}$.

Theorem 12.5. *Let ρ be an arbitrary type and τ a discrete type. Then $\mathsf{ips}_{\rho,\tau}$ is primitive recursively definable from $\mathsf{BBC}_{\bar{\rho}^{\mathbb{N}},\tau}$, provably in $\mathsf{Cont} + \mathsf{QF\text{-}BI}$.*

Proof. Suppose we are given $\varepsilon \colon \mathbb{N} \to J_\tau \rho$ and $q \colon \rho^{\mathbb{N}} \to \tau$. First, define $\tilde{q} \colon (\bar{\rho}^{\mathbb{N}})^{\mathbb{N}} \to \tau$ by

$$\tilde{q}(\gamma) := q(d(\gamma))$$

where the map $d \colon (\bar{\rho}^{\mathbb{N}})^{\mathbb{N}} \to \rho^{\mathbb{N}}$ is defined by

$$d(\gamma)(n) \overset{\rho}{:=} \begin{cases} \gamma_m[n] & \text{for least } m \leq n \text{ with } n \in \mathrm{dom}(\gamma_m) \\ 0_\rho & \text{if no such } m \text{ exists.} \end{cases}$$

Second, using the primitive recursor R_ρ define the selection function $\tilde{\varepsilon} \colon J_\tau \bar{\rho}^{\mathbb{N}}$ as

$$\tilde{\varepsilon}(p^{\bar{\rho}^{\mathbb{N}} \to \tau}) \overset{\bar{\rho}^{\mathbb{N}}}{:=} \lambda k \,.\, (1, \varepsilon_k(\lambda x^\rho \,.\, p(\{\tilde{t}_k * x\})))$$

where $\tilde{t}_0 = \langle\rangle$ and $\tilde{t}_k \overset{\rho^*}{:=} \langle \tilde{\varepsilon}(p)[0], \ldots, \tilde{\varepsilon}(p)[k-1] \rangle$. We extend this to a family of identical selection functions $\tilde{\varepsilon}_n := \tilde{\varepsilon}$. Now define

$$\Phi^{\varepsilon,q}(s^{\rho^*}) \overset{\rho^{\mathbb{N}}}{:=} \begin{cases} d(\mathsf{BBC}^{\tilde{\varepsilon},\tilde{q}}(\langle\rangle)) & \text{if } s = \langle\rangle \\ d(\mathsf{BBC}^{\tilde{\varepsilon},\tilde{q}}(\{\langle\{s\}\rangle\})) & \text{otherwise} \end{cases}$$

We claim that $\mathsf{ips}_{\rho,\tau}[\Phi]$. Note that to cut down on excessive syntax, given a sequence $\langle s_0^{\rho^*}, \ldots, s_{k-1} \rangle$ of finite sequences we write $\mathsf{BBC}(\langle s_0, \ldots, s_{k-1} \rangle)$ when we implicitly mean BBC applied to the partial array $\Gamma := \{\langle \{s_0\}, \ldots, \{s_{k-1}\} \rangle\} \colon \overline{(\bar{\rho}^{\mathbb{N}})}^{\mathbb{N}}$ defined by

$$\Gamma_m \overset{\mathbb{B} \times \bar{\rho}^{\mathbb{N}}}{=} \begin{cases} (1, \{s_m\}) & \text{if } m < k \\ (0, 0_{\bar{\rho}^{\mathbb{N}}}) & \text{otherwise.} \end{cases}$$

Now we prove the claim. For $s = \langle\rangle$ we have (suppressing parameters on Φ and BBC)

$$\begin{aligned} \Phi(\langle\rangle) &\overset{(*)}{=} \lambda n \,.\, \tilde{\varepsilon}_0(\underbrace{\lambda v \,.\, \tilde{q}(\mathsf{BBC}(\langle\rangle_0^v))}_{p})[n] \\ &= \lambda n \,.\, \varepsilon_n(\lambda x \,.\, \tilde{q}(\mathsf{BBC}(\langle\rangle_0^{\{\tilde{t}_n * x\}}))) \\ &= \lambda n \,.\, \varepsilon_n(\lambda x \,.\, q(d(\mathsf{BBC}(\langle \tilde{t}_n * x \rangle)))) \\ &\overset{(**)}{=} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, q(\Phi(t_n * x))) \end{aligned}$$

where $(**)$ we have $\tilde{t}_n = \langle \tilde{\varepsilon}_0(p)[0], \ldots, \tilde{\varepsilon}_0(p)[n-1] \rangle = [\Phi(\langle\rangle)](n) = t_n$. Note that $(*)$ follows because $n \in \mathrm{dom}(\mathsf{BBC}(\langle\rangle)_0)$ for all n and therefore $d(\mathsf{BBC}(\langle\rangle))(n) = \mathsf{BBC}(\langle\rangle)_0[n]$.

Now, for $|s| > 0$ we have

$$\begin{aligned}
\Phi(s) &\overset{(*)}{=} s \mathbin{@} \lambda n \,.\, \tilde{\varepsilon}_1(\underbrace{\lambda v \,.\, \tilde{q}(\mathsf{BBC}(\langle s\rangle_1^v))}_{p})[n] \\
&= s \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, \tilde{q}(\mathsf{BBC}(\langle s\rangle_1^{\{\tilde{t}_n * x\}}))) \\
&= s \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, q(d(\mathsf{BBC}(\langle s, \tilde{t}_n * x\rangle)))) \\
&\overset{(\dagger)}{=} s \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, q(d(\mathsf{BBC}(\langle s \mathbin{@} (\tilde{t}_n * x)\rangle)))) \\
&= s \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, q(\Phi(s \mathbin{@} (\tilde{t}_n * x)))) \\
&\overset{(**)}{=} s \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, q(\Phi(t_n * x)))
\end{aligned}$$

where for $(**)$ we have (for $n \geq |s|$)

$$\begin{aligned}
s \mathbin{@} (\tilde{t}_n * x) &= s \mathbin{@} (\langle \tilde{\varepsilon}_1(p)[0], \ldots, \tilde{\varepsilon}_1(p)[n-1]\rangle * x) \\
&= s * \langle \tilde{\varepsilon}_1(p)[|s|], \ldots, \tilde{\varepsilon}_1(p)[n-1]\rangle * x \\
&= [\Phi(s)](n) * x \\
&= t_n * x.
\end{aligned}$$

Note that $(*)$ follows because $n \in \mathrm{dom}(\mathsf{BBC}(\langle s\rangle)_0)$ iff $n < |s|$, but $n \in \mathrm{dom}(\mathsf{BBC}(\langle s\rangle)_1) = \mathrm{dom}(\tilde{\varepsilon}_1(\ldots))$ for all n. Therefore

$$d(\mathsf{BBC}(\langle s\rangle))(n) = \begin{cases} \{s\}[n] = s_n & \text{if } n < |s| \\ \tilde{\varepsilon}_1(p)[n] & \text{if } n \geq |s|. \end{cases}$$

Note that $\tilde{\varepsilon}_1(p)[n]$ is always included in the bounded search since $n \geq |s|$ implies $n \geq 1$. It remains to justify $(\dagger)$, which follows from the claim that whenever $0 < |s| < |t|$ we have

$$\tilde{q}(\mathsf{BBC}(\langle s, t\rangle)) = \tilde{q}(\mathsf{BBC}(\langle s \mathbin{@} t\rangle)).$$

Written informally, we're asserting that

$$\tilde{q}(\mathsf{BBC}\begin{pmatrix} & \vdots & \\ & \bot & \\ & t_j & \\ \vdots & \vdots & \\ \bot & t_{i+1} & \\ s_i & t_i & \\ \vdots & \vdots & \\ s_0 & t_0 & \ldots \end{pmatrix}) = \tilde{q}(\mathsf{BBC}\begin{pmatrix} \vdots & \\ \bot & \\ t_j & \\ \vdots & \\ t_{i+1} & \\ s_i & \\ \vdots & \\ s_0 & \ldots \end{pmatrix})$$

This intuitively follows because $\tilde{q}$ never looks at the elements below the cut-off diagonal d, so in particular ignores $\langle t_0, \ldots, t_i\rangle$. We now prove the claim, using bar induction on sequences $u \colon (\rho^*)^*$ relativised to those consisting only of non-empty sequences i.e. $S(u) :\equiv$

$\forall i < |u|(|u_i| > 0)$. Given such a sequence, define $u^\bullet, u^\circ \colon (\rho^*)^*$ with $|u^\bullet| = |u^\circ| = |u|$ primitive recursively in BBC as

$$u_i^\bullet \overset{\rho^*}{:=} [\pi_1 \tilde{\varepsilon}(\lambda v \,.\, \tilde{q}(\mathsf{BBC}(\langle s, t, u_0^\bullet, \dots, u_{i-1}^\bullet, v\rangle)))](k_i) * u_i$$
$$u_i^\circ \overset{\rho^*}{:=} [\pi_1 \tilde{\varepsilon}(\lambda v \,.\, \tilde{q}(\mathsf{BBC}(\langle s \mathbin{@} t, u_0^\circ, \dots, u_{i-1}^\circ, v\rangle)))](k_i) * u_i$$

where as before we implicitly mean *embeddings* in $\bar{\rho}^{\mathbb{N}}$ of s, t, $u_i^\bullet$, u_i° (and in turn $\langle s, t, u_0^\bullet, \dots, u_{i-1}^\bullet, v\rangle$ is embedded as a partial sequence over $\bar{\rho}^{\mathbb{N}}$) in the arguments of BBC, and $k_i := |t| + \sum_{j=0}^{i-1} |u_j|$ for $i < |u|$. The purpose of these constructions is actually to simplify the bar induction: The first k_i entries of $u_i^\bullet$ (similarly u_i°) represent 'dummy' values that would be computed by $\mathsf{BBC}(\langle s, t, u_0^\bullet, \dots, u_{i-1}^\bullet, v\rangle)$ but never seen by the outcome functional $\tilde{q}$, and so should not affect the value of $\tilde{q}(\mathsf{BBC}(\langle s, t\rangle * u^\bullet))$. We carry out bar induction on the quantifier-free predicate

$$P(u) :\equiv \tilde{q}(\mathsf{BBC}(\langle s, t\rangle * u^\bullet)) = \tilde{q}(\mathsf{BBC}(\langle s \mathbin{@} t\rangle * u^\circ)).$$

which we can informally visualise as

$$\tilde{q}(\mathsf{BBC}\left(\begin{array}{cccccc}
 & & & & & \vdots \\
 & & & & & u_{n-1,l_n-1} \\
 & & & & & \vdots \\
 & & & & & u_{n-1,0} \\
 & & & & & \bullet \\
 & & & & \vdots & \vdots \\
 & & & u_{1,l_1-1} & & \\
 & & & \vdots & & \\
 & & & u_{1,0} & & \\
 & & u_{0,l_0-1} & \bullet & & \\
 & \vdots & \vdots & \vdots & & \\
 & \bot & u_{0,0} & & & \\
 & t_j & \bullet & & & \\
\vdots & \vdots & \vdots & & & \\
\bot & t_{i+1} & & & & \\
s_i & t_i & & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
s_0 & t_0 & \underbrace{\bullet}_{u_0^\bullet} & \underbrace{\bullet}_{u_1^\bullet} & \cdots & \underbrace{\bullet}_{u_{n-1}^\bullet}
\end{array}\right)) = \tilde{q}(\mathsf{BBC}\left(\begin{array}{ccccc}
 & & & & \vdots \\
 & & & & u_{n-1,l_n-1} \\
 & & & & \vdots \\
 & & & & u_{n-1,0} \\
 & & & & \circ \\
 & & & \vdots & \vdots \\
 & & u_{1,l_1-1} & & \\
 & & \vdots & & \\
 & & u_{1,0} & & \\
 & u_{0,l_0-1} & \circ & & \\
\vdots & \vdots & \vdots & & \\
\bot & u_{0,0} & & & \\
t_j & \circ & & & \\
\vdots & \vdots & & & \\
t_{i+1} & & & & \\
s_i & & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
s_0 & \underbrace{\circ}_{u_0^\circ} & \underbrace{\circ}_{u_1^\circ} & \cdots & \underbrace{\circ}_{u_{n-1}^\circ}
\end{array}\right))$$

The claim then follows from $P(\langle\rangle)$ since $\langle\rangle^\bullet = \langle\rangle^\circ = \langle\rangle$. First, to establish the bar, suppose that $\alpha \colon (\rho^*)^{\mathbb{N}}$ is an infinite sequence of non-empty sequences, and let N be the point of continuity of $q_{s \mathbin{@} t}$ on $\tilde{\alpha} \colon \rho^{\mathbb{N}}$ defined by

$$\tilde{\alpha}(n) {=} \alpha_i(n - l_i) \text{ for least } i \leq n (l_i \leq n < l_{i+1}) \tag{12.1}$$

for $l_0 = 0$ and $l_i := \sum_{j=0}^{i-1} |\alpha_j|$. Note that $\tilde{\alpha}$ is just $\alpha_0 * \alpha_1 * \alpha_2 \ldots$, and because$|\alpha_i| > 0$ for all i we must have $[\tilde{\alpha}](N) \prec \alpha_0 * \alpha_1 * \ldots * \alpha_{N-1}$. Now,

$$d(\underbrace{\mathsf{BBC}(\langle s, t\rangle * [\alpha](N)^\bullet)}_{\gamma}) \overset{\rho^{\mathbb{N}}}{=} (s \mathbin{@} t) * \alpha_0 * \ldots * \alpha_{N-1} \mathbin{@} \ldots.$$

This follows because $n \in \mathrm{dom}(\gamma_0) \leftrightarrow n < |s|$, $n \in \mathrm{dom}(\gamma_1) \leftrightarrow n < |t|$ and for $i < N$, $n \in \mathrm{dom}(\gamma_{i+2}) \leftrightarrow n \in \mathrm{dom}(\{[\alpha](N)^\bullet_i\}) \leftrightarrow n < k_i + |\alpha_i| = |t| + l_{i+1}$, and therefore (for $k_i = |t| + l_i$)

$$d(\gamma)(n) = \begin{cases} \gamma_0[n] = s_n & \text{if } n < |s| \\ \gamma_1[n] = t_n & \text{if } |s| \leq n < |t| \\ \gamma_{i+2}[n] = \alpha_i(n - l_i - |t|) & \text{if } i < N \wedge k_i \leq n < k_{i+1} \\ \ldots & \text{otherwise} \end{cases}$$

(note that we always have $i + 2 \leq |t| + l_i < n$ so γ_{i+2} is included in the bounded search performed by d). By an analogous argument we see that

$$d(\mathsf{BBC}(\langle s \mathbin{@} t\rangle * [\alpha](N)^\circ)) = (s \mathbin{@} t) * \alpha_0 * \ldots * \alpha_{N-1} \mathbin{@} \ldots$$

and therefore by continuity

$$\tilde{q}(\mathsf{BBC}(\langle s, t\rangle * [\alpha](N)^\bullet)) = q_{s \mathbin{@} t}([\tilde{\alpha}](N) \mathbin{@} \ldots) = \tilde{q}(\mathsf{BBC}(\langle s \mathbin{@} t\rangle * [\alpha](N)^\circ)),$$

which establishes $P([\alpha](N))$, and therefore we have $\forall \alpha \in S \exists n P([\alpha](n))$.

For the induction step we prove $P(u)$ assuming that each u_i is non-empty. Setting $k_i = |t| + l_i = |t| + \sum_{j=0}^{i-1} |u_j|$ and $m = |u|$, as before we have

$$d(\underbrace{\mathsf{BBC}(\langle s, t\rangle * u^\bullet)}_{\gamma}) = (s \mathbin{@} t) * u_0 * \ldots * u_{m-1} * \beta$$

where $\beta(n) = \gamma_{m+2}(k_m + n)$. This follows as before because $n \in \mathrm{dom}(\gamma_0) \leftrightarrow n < |s|$, $n \in \mathrm{dom}(\gamma_1) \leftrightarrow n < |t|$, and for $i < m$, $n \in \mathrm{dom}(\gamma_{i+2}) \leftrightarrow n \in \mathrm{dom}(u_i^\bullet) \leftrightarrow n < k_i + |u_i| = k_{i+1}$. Finally, we have $n \in \mathrm{dom}(\gamma_{m+2}) = \mathrm{dom}(\tilde{\varepsilon}_{m+2}(\ldots))$ for all n, therefore

$$d(\gamma)(n) = \begin{cases} \gamma_0[n] = s_n & \text{if } n < |s| \\ \gamma_1[n] = t_n & \text{if } |s| \leq n < |t| \\ \gamma_{i+2}[n] = u_i(n - k_i) & \text{if } i < m \wedge k_i \leq n < k_{i+1} \\ \gamma_{m+2}[n] = \beta(n - k_m) & \text{if } k_m \leq n. \end{cases}$$

Now,

$$\begin{aligned} \beta(n) &= \tilde{\varepsilon}_{m+2}(\underbrace{\lambda v \,.\, \tilde{q}(\mathsf{BBC}(((\langle s, t\rangle * u^\bullet)^v_{m+2}))}_{p})[k_m + n] \\ &= \varepsilon_{k_m+n}(\lambda x \,.\, \tilde{q}(\mathsf{BBC}(((\langle s, t\rangle * u^\bullet * \langle \tilde{t}_{k_m+n} *_\rho x\rangle)))) \\ &\overset{(+)}{=} \varepsilon_{k_m+n}(\lambda x \,.\, \tilde{q}(\mathsf{BBC}(((\langle s, t\rangle * (u * \langle [\beta](n) *_\rho x\rangle)^\bullet))). \end{aligned}$$

Note that in the latter two formulas concatenation of type ρ^* is not to be confused with concatenation of type ρ, which we carefully highlight. For $(+)$ we clearly have $u^\bullet_i = (u {*} \langle [\beta](n) {*_\rho} x\rangle)^\bullet_i$ for $i < m$ since the operation $(-)^\bullet$ is carried out recursively on sequences, and

$$\begin{aligned}\tilde{t}_{k_m+n} * x &= \langle \tilde{\varepsilon}(p)[0], \ldots, \tilde{\varepsilon}(p)[k_m - 1]\rangle * \langle \tilde{\varepsilon}(p)[k_m], \ldots, \tilde{\varepsilon}(p)[k_m + n - 1]\rangle * x \\ &= [\pi_1 \tilde{\varepsilon}(\lambda v \mathrel{.} \tilde{q}(\mathsf{BBC}(\langle s, t, u^\bullet_0, \ldots, u^\bullet_{m-1}, v\rangle)))](k_m) * [\beta](n) * x \\ &= (u * \langle [\beta](n) *_\rho x\rangle)^\bullet_m\end{aligned}$$

By an analogous argument we have

$$d(\mathsf{BBC}(\langle s \mathbin{@} t\rangle * u^\circ)) = (s \mathbin{@} t) * u_0 * \ldots * u_{m-1} * \beta'$$

where $\beta'(n) = \varepsilon_{k_m+n}(\lambda x \mathrel{.} \tilde{q}(\mathsf{BBC}(\langle s \mathbin{@} t\rangle * (u * \langle [\beta'](n) *_\rho x\rangle)^\circ))$. We can now use the bar induction hypothesis to prove that $\beta(n) = \beta'(n)$ for all n. Assume by normal induction that $\beta(m) = \beta'(m)$ for $m < n$. Then

$$\begin{aligned}\beta(n) &= \varepsilon_{k_m+n}(\lambda x \mathrel{.} \tilde{q}(\mathsf{BBC}((\langle s, t\rangle * (u * \langle [\beta](n) *_\rho x\rangle)^\bullet))) \\ &\overset{B.I.H.}{=} \varepsilon_{k_m+n}(\lambda x \mathrel{.} \tilde{q}(\mathsf{BBC}((\langle s \mathbin{@} t\rangle * (u * \langle [\beta](n) *_\rho x\rangle)^\circ))) \\ &\overset{I.H.}{=} \varepsilon_{k_m+n}(\lambda x \mathrel{.} \tilde{q}(\mathsf{BBC}((\langle s \mathbin{@} t\rangle * (u * \langle [\beta'](n) *_\rho x\rangle)^\circ))) \\ &= \beta'(n).\end{aligned}$$

Finally, then, we have

$$\begin{aligned}\tilde{q}(\mathsf{BBC}(\langle s, t\rangle * u^\bullet)) &= q((s \mathbin{@} t) * u_0 * \ldots u_{m-1} * \beta) \\ &\overset{\beta=\beta'}{=} q((s \mathbin{@} t) * u_0 * \ldots * u_{m-1} * \beta') \\ &= \tilde{q}(\mathsf{BBC}(\langle s \mathbin{@} t\rangle * u^\circ))\end{aligned}$$

and we complete the induction step, and hence the proof. □

Remark 12.6. We conjecture that the above instance of BBC naturally defines a variant of ips_0 of ips which is augmented to include boolean markers which indicate whether or not points are being 'tested' by selection functions, given by

$$\mathsf{ips}_0^{\varepsilon,q}(\langle b, s\rangle^{(\mathbb{B}\times\rho)^*}) \overset{(\mathbb{B}\times\rho)^{\mathbb{N}}}{:=} \langle b, s\rangle \mathbin{@} \mathsf{ips}_0(\langle b, s\rangle * \langle 1, a_s\rangle)$$

where $a_s := \varepsilon_{|s|}(\lambda x \mathrel{.} \bar{q}(\mathsf{ips}_0(\langle b, s\rangle * \langle 0, x\rangle)))$, $\bar{q}(\alpha) := q(\pi_1 \alpha)$ and the parameters have type $\varepsilon_n \colon J_\tau \rho$, $q \colon \rho^{\mathbb{N}} \to \tau$. The idea is that the 0-marker would coincide with a 'shift' to the column on the right. Given a sequence $\langle b, s\rangle$ with zeroes $b_{n_0} = \ldots = b_{n_{m-1}} = 0$, and assuming w.l.o.g. that the last element of $\langle b, s\rangle$ has marker 0, we claim the one could define

$$\mathsf{ips}_0^{\varepsilon,q}(\langle b, s\rangle) = \mathsf{BBC}^{\tilde{\varepsilon},\tilde{q}}(u^\bullet)$$

where $u_0 := \langle s_0, \ldots, s_{n_0} \rangle$, $u_i := \langle s_{n_i+1}, \ldots, s_{n_{i+1}} \rangle$ and $u^\bullet$ is defined as in the preceding proof (but without the initial entries s and t). Then, the bar induction (and reliance of Cont) would simply be needed to show that the value of $\bar{q}(\mathsf{ips}_0(\langle b, s \rangle))$ is independent of the markers b (and therefore ips_0 defines ips), and thus one could adapt our proof to show that BBC defines ips_0 in $\mathsf{E\text{-}HA}^\omega$ alone.

Theorem 12.7. IPS*,* OR*,* UR *and* BBC *are all primitive recursively equivalent over* $\mathsf{Cont}+$ $\mathsf{QF\text{-}BI}$*.*

Proof. We have the following chain of definability results:

$$\mathsf{IPS} \overset{\text{Thm. 12.1}}{\geq} \mathsf{OR} \overset{\text{Prop. 9.20}}{\geq} \mathsf{UR} \overset{\text{Prop. 9.16}}{\geq} \mathsf{BBC} \overset{\text{Thm. 12.5}}{\geq} \mathsf{ips} \overset{\text{Prop. 12.4}}{\geq} \mathsf{IPS}.$$

□

Remark 12.8. In fact, by inspection one will notice that each of these results is true instance-wise, so IPS, OR, UR and BBC are also instance-wise equivalent.

APPENDIX

A

Omitted Proofs

Proposition A.1. *Let* IPS *be defined by*

$$\mathsf{IPS}^{\varepsilon,q}_{\rho,\tau}(s^{\rho^*}) \overset{\rho^{\mathbb{N}}}{:=} s \mathbin{@} \mathsf{IPS}^{\varepsilon,q}_{\rho,\tau}(s * a_s)$$

where $a_s := \varepsilon_s(\lambda x \mathbin{.} q(\mathsf{IPS}(s * x)))$ *and* $\widetilde{\mathsf{IPS}}$ *be defined by*

$$\widetilde{\mathsf{IPS}}^{\varepsilon,q}(s) := s \mathbin{@} \lambda n \mathbin{.} \varepsilon_{t_n}(\lambda x \mathbin{.} q(\widetilde{\mathsf{IPS}}^{\varepsilon,q}(t_n * x)))$$

where $t_n := [\widetilde{\mathsf{IPS}}(s)](n)$*. Then* $\mathsf{E\text{-}HA}^{\omega} + \mathsf{IPS}[\Phi] \vdash \widetilde{\mathsf{IPS}}[\Phi]$ *and* $\mathsf{E\text{-}HA}^{\omega} + \widetilde{\mathsf{IPS}}[\Phi] \vdash \mathsf{IPS}[\Phi]$*.*

Proof. We first show that $\widetilde{\mathsf{IPS}}[\mathsf{IPS}]$. By definition $\mathsf{IPS}^{\varepsilon,q}(s)(n) = s_n$ for all $n < |s|$, and for $n \geq |s|$ we have

$$\begin{aligned}
\mathsf{IPS}(s)(n) &\overset{L.9.8}{=} \mathsf{IPS}(t_n)(n) \\
&= (t_n \mathbin{@} \mathsf{IPS}(t_n * a_{t_n}))(n) \\
&= a_{t_n} \\
&= \varepsilon_{t_n}(\lambda x \mathbin{.} q(\mathsf{IPS}(t_n * x))).
\end{aligned}$$

Conversely, to show that $\mathsf{IPS}[\widetilde{\mathsf{IPS}}]$, we use course of values induction on $t_n = \tilde{t}_n$ where $t_n = [\widetilde{\mathsf{IPS}}(s)](n)$ and $\tilde{t}_n = [\widetilde{\mathsf{IPS}}(s * a_s)](n)$, and then the result immediately follows. Clearly this is true for $n = |s|$. Now

$$\widetilde{\mathsf{IPS}}(s)(|s|) = \varepsilon_s(\lambda x \mathbin{.} q(\widetilde{\mathsf{IPS}}(s * x))) = a_s = \widetilde{\mathsf{IPS}}(s * a_s)(|s|)$$

and therefore $t_{|s|+1} = \tilde{t}_{|s|+1}$. For the induction step, assume that $t_n = \tilde{t}_n$ for $n \geq |s| + 1$. Then

$$\widetilde{\mathsf{IPS}}(s)(n) = \varepsilon_{t_n}(\lambda x \mathbin{.} q(\widetilde{\mathsf{IPS}}(t_n * x))) \overset{I.H.}{=} \varepsilon_{\tilde{t}_n}(\lambda x \mathbin{.} q(\widetilde{\mathsf{IPS}}(\tilde{t}_n * x))) = \widetilde{\mathsf{IPS}}(s * a_s)(n)$$

and therefore $t_{n+1} = t_n * \widetilde{\mathsf{IPS}}(s)(n) \overset{I.H.}{=} \tilde{t}_n * \widetilde{\mathsf{IPS}}(s * a_s)(n) = \tilde{t}_{n+1}$, which completes the induction, and the result follows. □

Equivalence of our products of selection functions and those of Escardó and Oliva

In much of the existing literature [28, 29, 30] products of selection functions are constructed slightly differently to ours. In particular, the implicit product of selection functions is defined as

$$\mathsf{sIPS}^{\varepsilon}_{\rho,\tau}(q)(s) \overset{\rho^{\mathbb{N}}}{:=} \lambda n\,.\,\varepsilon_{s*t_n}(\lambda x\,.\,q_{t_n*x}(\mathsf{sIPS}^{\varepsilon}(q_{t_n*x})(s*t_n*x))),$$

where $t_n := [\mathsf{sIPS}^{\varepsilon}(q)(s)](n)$ and $q_s(\alpha) := q(s*\alpha)$, while the simple product is given by

$$\mathsf{sips}^{\varepsilon}_{\rho,\tau}(q)(m) \overset{\rho^{\mathbb{N}}}{:=} \lambda n\,.\,\varepsilon_{m+n}(\lambda x\,.\,q_{t_n*x}(\mathsf{sips}^{\varepsilon}(q_{t_n*x})(m+n+1))).$$

where $t_n := [\mathsf{sips}^{\varepsilon}(q)(m)](n)$. Here there are two essential differences, namely that the function q updates with computes values, and only the tail end of the product is given as output. However, these versions are easily (though rather tediously) shown to be equivalent to our formulation in the continuous functionals.

Proposition A.2. IPS*,* sIPS*,* ips *and* sips *are all primitive recursively equivalent over* $\mathsf{Cont}+\mathsf{QF\text{-}BI}$.

Proof. In [28, Theorem 4.5] it is proved that $\mathsf{sIPS} =_{\mathsf{T}} \mathsf{sips}$ over $\mathsf{E\text{-}HA}^{\omega}$. To complete the proof we show that $\mathsf{IPS} \geq_{\mathsf{T}} \mathsf{ips}$, $\mathsf{sIPS} \geq_{\mathsf{T}} \mathsf{IPS}$ and $\mathsf{ips} \geq_{\mathsf{T}} \mathsf{sips}$. The first of these is trivial, and the second is not much more difficult. Defining $\Phi^{\varepsilon,q}(s) := s*\mathsf{sIPS}^{\varepsilon}_{\rho,\tau}(q_s)(s)$ we see that $\mathsf{IPS}_{\rho,\tau}[\Phi]$: For $n<|s|$ we have $\Phi(s)(k)=s_k$, and for $n=|s|+k$ we have

$$\begin{aligned}
\Phi(s)(n) &= s*\mathsf{sIPS}(q_s)(s)(|s|+k)\\
&= \varepsilon_{s*t_k}(\lambda x\,.\,(q_s)_{t_n*x}(\mathsf{sIPS}((q_s)_{t_k*x})(s*t_k*x)))\\
&= \varepsilon_{s*t_k}(\lambda x\,.\,q(s*t_n*x*\mathsf{sIPS}(q_{s*t_k*x})(s*t_k*x)))\\
&= \varepsilon_{s*t_k}(\lambda x\,.\,q(\Phi(s*t_k*x)))\\
&\overset{(*)}{=} \varepsilon_{\tilde{t}_n}(\lambda x\,.\,q(\Phi(\tilde{t}_n*x)))
\end{aligned}$$

where for $(*)$ we have

$$s*t_k = s*[\mathsf{sIPS}(q_s)(s)](k) = [\Phi(s)](|s|+k) = \tilde{t}_n.$$

To show $\mathsf{ips} \geq \mathsf{sips}$ requires a little more effort, as seems to always be the case going between bar recursion where computed values remain as parameters (as in ips) and bar recursion where computes values are encoded in the outcome function (as in sips). We define $\Phi^{\varepsilon}(q)(m) := \mathsf{tail}_m(\mathsf{ips}^{\varepsilon,q^m}_{\rho,\tau}(0_m))$ where $q^m(\alpha) := q(\mathsf{tail}_m(\alpha))$ and $0_m := [0_{\rho^{\mathbb{N}}}](m)$.

Then $\mathsf{sips}_{\rho,\tau}[\Phi]$:

$$\begin{aligned}
\Phi(q)(m)(n) &= \mathsf{tail}_m(\mathsf{ips}^{q^m}(0_m))(n)\\
&= \mathsf{ips}^{q^m}(0_m)(m+n)\\
&= \varepsilon_{m+n}(\lambda x\ .\ q^m(\mathsf{ips}^{q^m}(t_{m+n} * x)))\\
&= \varepsilon_{m+n}(\lambda x\ .\ q^m(t_{m+n} * x * \mathsf{tail}_{m+n+1}(\mathsf{ips}^{q^m}(t_{m+n} * x))))\\
&\overset{(*)}{=} \varepsilon_{m+n}(\lambda x\ .\ q^m(0_m * \tilde{t}_n * x * \mathsf{tail}_{m+n+1}(\mathsf{ips}^{q^m}(0_m * \tilde{t}_n * x))))\\
&= \varepsilon_{m+n}(\lambda x\ .\ q_{\tilde{t}_n * x}(\mathsf{tail}_{m+n+1}(\mathsf{ips}^{q^m}(0_m * \tilde{t}_n * x))))\\
&\overset{(+)}{=} \varepsilon_{m+n}(\lambda x\ .\ q_{\tilde{t}_n * x}(\mathsf{tail}_{m+n+1}(\mathsf{ips}^{(q_{\tilde{t}_n * x})^{m+n+1}}(0_{m+n+1}))))\\
&= \varepsilon_{m+n}(\lambda x\ .\ q_{\tilde{t}_n * x}(\Phi(q_{\tilde{t}_n * x})(m+n+1)))
\end{aligned}$$

where for $(*)$ we have

$$t_{m+n} = [\mathsf{ips}^{q^m}(0_m)](m+n) = 0_m * [\mathsf{tail}_m(\mathsf{ips}^{q^m}(0_m))](n) = 0_m * [\Phi(q)(m)](n) = 0_m * \tilde{t}_m$$

and $(+)$ follows from $P(\langle\rangle)$ which is proved using $\mathsf{Cont} + \mathsf{QF\text{-}BI}$, where

$$P(r) :\equiv q_t(\mathsf{tail}_{m+n+1}(\mathsf{ips}^{q^m}(0_m * t * r))) = q_t(\mathsf{tail}_{m+n+1}(\mathsf{ips}^{(q_t)^{m+n+1}}(0_{m+n+1} * r)))$$

for $t := \tilde{t}_n * x$. Let N be the point of continuity of q_t on α. Then

$$\begin{aligned}
q_t(\mathsf{tail}_{m+n+1}(\mathsf{ips}^{q^m}(0_m * t * [\alpha](N)))) &= q_t([\alpha](N) \mathbin{@} \dots)\\
&= q_t(\mathsf{tail}_{m+n+1}(\mathsf{ips}^{(q_t)^{m+n+1}}(0_{m+n+1} * [\alpha](N))))
\end{aligned}$$

which establishes the bar, and assuming $\forall y P(r * y)$ we have (using the alternative formulation of ips)

$$\begin{aligned}
q_t(\mathsf{tail}_{m+n+1}(\mathsf{ips}^{q^m}(0_m * t * r))) &= q_t(\mathsf{tail}_{m+n+1}(\mathsf{ips}^{q^m}(0_m * t * r * a)))\\
&\overset{B.I.H.}{=} q_t(\mathsf{tail}_{m+n+1}(\mathsf{ips}^{(q_t)^{m+n+1}}(0_{m+n+1} * r * a)))\\
&\overset{(*)}{=} q_t(\mathsf{tail}_{m+n+1}(\mathsf{ips}^{(q_t)^{m+n+1}}(0_{m+n+1} * r * \tilde{a})))\\
&= q_t(\mathsf{tail}_{m+n+1}(\mathsf{ips}^{(q_t)^{m+n+1}}(0_{m+n+1} * r)))
\end{aligned}$$

where for $(*)$ we have

$$\begin{aligned}
a &= \varepsilon_{m+n+1+|r|}(\lambda y\ .\ q^m(\mathsf{ips}^{q^m}(0_m * t * r * y))\\
&= \varepsilon_{m+n+1+|r|}(\lambda y\ .\ q_t(\mathsf{tail}_{m+n+1}(\mathsf{ips}^{q^m}(0_m * t * r * y)))\\
&\overset{B.I.H.}{=} \varepsilon_{m+n+1+|r|}(\lambda y\ .\ q_t(\mathsf{tail}_{m+n+1}(\mathsf{ips}^{(q_t)^{m+n+1}}(0_{m+n+1} * r * y))))\\
&= \varepsilon_{m+n+1+|r|}(\lambda y\ .\ (q_t)^{m+n+1}(\mathsf{ips}^{(q_t)^{m+n+1}}(0_{m+n+1} * r * y)))\\
&= \tilde{a}.
\end{aligned}$$

□

Bibliography

[1] F. Aschieri. *Learning, Realizability and Games in Classical Arithmetic*. PhD thesis, Universita degli Studi di Torino and Queen Mary, University of London, 2011.

[2] F. Aschieri. Transfinite update procedures for predicative systems of analysis. In *Computer Science Logic (CSL'11)*, volume 12 of *Leibniz International Proceedings in Informatics*, pages 1–15, 2011.

[3] F. Aschieri and S. Berardi. Interactive learning-based realizability for Heyting arithmetic with EM1. *Logical Methods in Computer Science*, 6(3), 2010.

[4] J. Avigad. Update procedures and the 1-consistency of arithmetic. *Mathematical Logic Quarterly*, 48(1):3–13, 2002.

[5] J. Avigad and S. Feferman. Gödel's functional ("Dialectica") interpretation. In S. R. Buss, editor, *Handbook of Proof Theory*, volume 137 of *Studies in Logic and the Foundations of Mathematics*, pages 337–405. North Holland, Amsterdam, 1998.

[6] J. Avigad, P. Gerhardy, and H. Towsner. Local stability of ergodic averages. *Transactions of the American Mathematical Society*, 362:261–288, 2010.

[7] G. Bellin. Ramsey interpreted: a parametric version of Ramsey's theorem. In *Logic and computation (Pittsburgh, PA, 1987)*, volume 106, pages 17–37. Amer. Math. Soc., Providence, RI, 1990.

[8] S. Berardi, M. Bezem, and T. Coquand. On the computational content of the axiom of choice. *Journal of Symbolic Logic*, 63(2):600–622, 1998.

[9] U. Berger. The Berardi-Bezem-Coquand functional in a domain-theoretic setting. Unpublished note, available from author's webpage, 2002.

[10] U. Berger. A computational interpretation of open induction. In F. Titsworth, editor, *Proceedings of the Nineteenth Annual IEEE Symposium on Logic in Computer Science*, pages 326–334. IEEE Computer Society, 2004.

[11] U. Berger and P. Oliva. Modified bar recursion and classical dependent choice. *Lecture Notes in Logic*, 20:89–107, 2005.

[12] U. Berger and P. Oliva. Modified bar recursion. *Mathematical Structures in Computer Science*, 16(2):163–183, 2006.

[13] U. Berger and M. Seisenberger. Applications of inductive definitions and choice principles to program synthesis. In *From Sets and Types to Topology and Analysis Towards Practicable Foundations for Constructive Mathematics*, volume 48 of *Oxford Logic Guides*, pages 137–148. Oxford University Press, 2005.

[14] M. Bezem. Strong normalization of bar recursive terms without using infinite terms. *Arch. Math. Log. Grundlagenforsch*, 25:175–181, 1985.

[15] M. Bezem. Strongly majorizable functionals of finite type: A model for bar recursion containing discontinuous functionals. *Journal of Symbolic Logic*, 50:652–660, 1985.

[16] M. Bezem. Equivalence of bar recursors in the theory of functionals of finite type. *Archive for Mathematical Logic*, 27:149–160, 1988.

[17] W. Burr. *Functionals in Set Theory and Arithmetic.* PhD thesis, Westfälische Wilhelms-Universität Münster, 1998.

[18] S. R. Buss. First-order proof theory of arithmetic. In S. R. Buss, editor, *Handbook of Proof Theory*, volume 137 of *Studies in Logic and the Foundations of Mathematics*, pages 79–147. North Holland, Amsterdam, 1998.

[19] E. A. Cichon and E. Tahhan-Bittar. Ordinal recursive bounds for Higman's theorem. *Theoretical Computer Science*, 201(1–2):63–84, 1998.

[20] T. Coquand. Constructive topology and combinatorics. In *Constructivity in Computer Science*, volume 613 of *LNCS*, pages 159–164, 1991.

[21] T. Coquand. An analysis of Ramsey's theorem. *Information and Computation*, 110(2):297–304, 1994.

[22] T. Coquand. A semantics of evidence for classical arithmetic. *Journal of Symbolic Logic*, 60:325–337, 1995.

[23] T. Coquand and D. Fridlender. A proof of Higman's lemma by structural induction. Unpublished Manuscript, available from authors' webpage, 1993.

[24] V. de Paiva. *The Dialectica Categories.* PhD thesis, University of Cambridge, 1991. Published as Technical Report 213, Computer Laboratory, University of Cambridge.

[25] P. Erdős, A. Hajnal, A. Máté, and R. Rado. *Combinatorial Set Theory: Partition Relations for Cardinals*, volume 106 of *Studies in Logic and the Foundations of Mathematics.* North-Holland Publishing Company, 1984.

[26] Y. L. Ershov. Model C of partial continuous functionals. In *Logic Colloquium*, pages 455–467. North Holland, Amsterdam, 1977.

[27] M. Escardó. Exhaustible sets in higher-type computation. *Logical Methods in Computer Science*, 4(3):1–37, 2008.

[28] M. Escardó and P. Oliva. Bar recursion and products of selection functions. Preprint, available from authors' webpage, 2010.

[29] M. Escardó and P. Oliva. Computational interpretation of analysis via the products of selection functions. In *Proceedings of CiE 2010*, volume 6158 of *LNCS*, pages 141–150. 2010.

[30] M. Escardó and P. Oliva. Selection functions, bar recursion and backward induction. *Mathematical Structures in Computer Science*, 20(2):127–168, 2010.

[31] M. Escardó and P. Oliva. Sequential games and optimal strategies. *Royal Society Proceedings A*, 467:1519–1545, 2011.

[32] M. Escardó, P. Oliva, and T. Powell. System T and the product of selection functions. In M. Bezem, editor, *Computer Science Logic (CSL'11)*, volume 12 of *LIPIcs*, pages 233–247, 2011.

[33] G. Ferreira and P. Oliva. On the relation between various negative translations. In *Logic, Construction, Computation*, volume 3 of *Onton-Verlag Mathematical Logic*, pages 227–258. 2012.

[34] H. Friedman. Classical and intuitionistically provable recursive functions. In *Higher Set Theory*, volume 669 of *LNM*, pages 21–27. Springer, 1978.

[35] R. Gandy and M. Hyland. Computable and recursively countable functionals of higher type. In R. Gandy and M. Hyland, editors, *Logic Colloquium 1976*, pages 407–438. North-Holland, Amsterdam, 1977.

[36] J. Gaspar and U. Kohlenbach. On Tao's "finitary" infinite pigeonhole principle. *Journal of Symbolic Logic*, 75(1):355–371, 2010.

[37] K. Gödel. Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes. *dialectica*, 12:280–287, 1958.

[38] G. Higman. Ordering by divisibility in abstract algebras. *Proceedings of the London Mathematical Society*, 2:326–336, 1952.

[39] W. A. Howard. Functional interpretation of bar induction by bar recursion. *Compositio Mathematica*, 20:107–124, 1968.

[40] W. A. Howard. Ordinal analysis of bar recursion of type zero. *Compositio Mathematica*, 42:105–119, 1981.

[41] W. A. Howard. Ordinal analysis of simple cases of bar recursion. *Journal of Symbolic Logic*, 46:17–31, 1981.

[42] W. A. Howard and G. Kreisel. Transfinite induction and bar induction of typesnzero and one, and the role of continuity in intuitionistic analysis. *Journal of Symbolic Logic*, 31:325–358, 1966.

[43] J. M. E. Hyland. Filter spaces and continuous functionals. *Annals of Mathematical Logic*, 16:101–143, 1979.

[44] D. Ilik and K. Nakata. A direct constructive proof of open induction on cantor space. Preprint, available at http://arxiv.org/abs/1209.2229, 2012.

[45] S. C. Kleene. Countable functionals. In A. Heyting, editor, *Constructivity in Mathematics*, pages 81–100. North-Holland, Amsterdam, 1959.

[46] U. Kohlenbach. *Theory of Majorizable and Continuous Functionals and their Use for the Extraction of Bounds from Non-Constructive Proofs: Effective Moduli of Uniqueness for Best Approximations from Ineffective Proofs of Uniqueness (German).* PhD thesis, Frankfurt, 1990.

[47] U. Kohlenbach. On the no-counterexample interpretation. *Journal of Symbolic Logic*, 64:1491–1511, 1999.

[48] U. Kohlenbach. *Applied Proof Theory: Proof Interpretations and their Use in Mathematics.* Monographs in Mathematics. Springer, 2008.

[49] U. Kohlenbach. Gödel functional interpretation and weak compactness. *Annals of Pure and Applied Logic*, 163:1560–1579, 2012.

[50] U. Kohlenbach. A uniform quantitative form of sequential weak compactness and Baillon's nonlinear ergodic theorem. *Communications in Contemporary Mathematics*, 14, 20 pp., 2012.

[51] U. Kohlenbach and P. Oliva. A systematic way of analyzing proofs in mathematics. *Proceedings of the Steklov Institute of Mathematics*, 242:136–164, 2003.

[52] A.N. Kolmogorov. On the principles of the excluded middle (Russian). *Matematicheskii Sbornik*, 32:646–667, 1925.

[53] G. Kreisel. On the interpretation of non-finitist proofs, Part I. *Journal of Symbolic Logic*, 16:241–267, 1951.

[54] G. Kreisel. Interpretation of analysis by means of functionals of finite type. In A. Heyting, editor, *Constructivity in Mathematics*, pages 101–128. North-Holland, Amsterdam, 1959.

[55] A. Kreuzer. *Proof mining and Combinatorics : Program Extraction for Ramsey's Theorem for Pairs*. PhD thesis, TU Darmstadt, 2012.

[56] A. Kreuzer. Der Satz von Ramsey für Paare und beweisbar rekursive Funktionen. Diploma thesis, February 2009.

[57] A. Kreuzer and U. Kohlenbach. Ramsey's theorem for pairs and provably recursive functions. *Notre Dame Journal of Formal Logic*, 50:427–444, 2009.

[58] A. Kreuzer and U. Kohlenbach. Term extraction and Ramsey's theorem for pairs. *Journal of Symbolic Logic*, 77(3):853–895, 2012.

[59] J. B. Kruskal. Well-quasi-ordering, the tree theorem, and Vázsonyi's conjecture. *Transactions of the American Mathematical Society*, 95:210–225, 1960.

[60] S. Kuroda. Intuitionistische Untersuchungen der formalistischen Logik. *Nagoya Mathematical Journal*, 3:35–47, 1951.

[61] H. Luckhardt. *Extensional Gödel Functional Interpretation*, volume 306 of *Springer Lecture Notes in Mathematics.* 1973.

[62] A. Marcone. On the logical strength of Nash-Williams' theorem on transfinite sequences. In W. Hodges, M. Hyland, C. Steinhorn, and J. Truss, editors, *Logic: From Foundations to Applications, Keele 1993*, Oxford Science Publications, pages 327–351. Oxford University Press, 1996.

[63] C. R. Murthy. *Extracting Constructive Content from Classical Proofs*. PhD thesis, Ithaca, New York, 1990.

[64] C. St. J. A. Nash-Williams. On well-quasi-ordering finite trees. *Proceedings of the Cambridge Philosophical Society*, 59:833–835, 1963.

[65] D. Normann. The continuous functionals. In S. Abramsky, S. Artemov, R. A. Shore, and A. S. Troelstra, editors, *Handbook of Computability Theory*, volume 140 of *Studies in Logic and the Foundations of Mathematics*, pages 251–275. North Holland, Amsterdam, 1999.

[66] P. Oliva. Understanding and using Spector's bar recursive interpretation of classical analysis. In A. Beckmann, U. Berger, B. Löwe, and J. V. Tucker, editors, *Proceedings of CiE'2006*, volume 3988 of *LNCS*, pages 423–234, 2006.

[67] P. Oliva. Unifying functional interpretations. *Notre Dame Journal of Formal Logic*, 47(2):263–290, 2006.

[68] P. Oliva and T. Powell. A constructive interpretation of Ramsey's theorem via the product of selection functions. *To appear: Mathematical Structures in Computer Science*, 2012. Preprint available at http://arxiv.org/abs/1204.5631.

[69] P. Oliva and T. Powell. A game-theoretic computational interpretation of proofs in classical analysis. *To appear: Gentzen Centenary volume*, 2012. Preprint available at http://arxiv.org/abs/1204.5244.

[70] P. Oliva and T. Powell. On Spector's bar recursion. *Mathematical Logic Quarterly*, 58:356–365, 2012.

[71] J. B. Paris and L. A. S. Kirby. Σ_n-collection schemas in arithmetic. In *Logic Colloquium '77*, pages 199–210. North Holland, Amserdam, 1978.

[72] C. Parsons. On a number theoretic choice schema and its relation to induction. In A. Kino, J. Myhill, and R. E. Vesley, editors, *Intuitionism and Proof Theory: Proceedings of the Summer Conference at Buffalo, N.Y. 1968*, pages 459–473. North Holland, Amserdam, 1970.

[73] C. Parsons. On n-quantifier induction. *The Journal of Symbolic Logic*, 37:466–482, 1972.

[74] T. Powell. Applying Gödel's Dialectica interpretation to obtain a constructive proof of Higman's lemma. In *Proceedings of Classical Logic and Computation*, volume 97 of *EPTCS*, pages 49–62, 2012.

[75] F. Ramsey. On a problem of formal logic. *Proceedings of the London Mathematical Society*, s2-30(1):264–286, 1930.

[76] J.-C. Raoult. Proving open properties by induction. *Information Processing Letters*, 29:19–23, 1988.

[77] P. Safarik. The interpretation of the Bolzano-Weierstrass principle using bar recursion. Master's thesis, TU Darmstadt, 2010.

[78] P. Safarik and U. Kohlenbach. On the interpretation of the Bolzano-Weierstrass principles. *Mathematical Logic Quarterly*, 56(5):508–532, 2010.

[79] B. Scarpellini. A model for bar recursion of higher types. *Compositio Mathematica*, 23:132–153, 1971.

[80] S. Schmitz and P. Schnoebelen. Multiply-recursive upper bounds with Higman's lemma. In *Automata, Languages and Programming*, volume 6756 of *LNCS*, pages 441–452. 2011.

[81] H. Schwichtenberg. Dialectica interpretation of well-founded induction. *Mathematical Logic Quarterly*, 54(3):229–239, 2008.

[82] M. Seisenberger. *On the Constructive Content of Proofs.* PhD thesis, LudwigMaximiliansUniversität München, 2003.

[83] W. Sieg. Fragments of arithmetic. *Annals of Pure and Applied Logic*, 28:33–71, 1985.

[84] S. G. Simpson. *Subsystems of Second Order Arithmetic.* Perspectives in Mathematical Logic. Springer, Berlin, 1999.

[85] E. Specker. Nicht konstruktiv beweisbare Sätze der Analysis. *Journal of Symbolic Logic*, 14:145–158, 1949.

[86] C. Spector. Provably recursive functionals of analysis: a consistency proof of analysis by an extension of principles in current intuitionistic mathematics. In F. D. E. Dekker, editor, *Recursive Function Theory: Proc. Symposia in Pure Mathematics*, volume 5, pages 1–27. American Mathematical Society, Providence, Rhode Island, 1962.

[87] W. W. Tait. Normal form theorem for bar recursive functions of finite type. In *Proceedings of 2nd Scandinavian Logic Symposium*, pages 353–367. North Holland, 1971.

[88] T. Tao. Soft analysis, hard analysis, and the finite convergence principle. Essay, published as Ch. 1.3 of [**?**], original version available online at http://terrytao.wordpress.com/2007/05/23/soft-analysis-hard-analysis-and-the-finite-convergence-principle/, 2008.

[89] A. Tarski, A. Mostowski, and R. M. Robinson. *Undecidable Theories.* North-Holland, Amsterdam, 1953.

[90] A. S. Troelstra. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*, volume 344 of *Lecture Notes in Mathematics.* Springer, Berlin, 1973.

[91] W. Veldman. An intuitionistic proof of Kruskal's theorem. *Archive for Mathematical Logic*, 43(2):215–264, 2004.

[92] W. Veldman. The principle of open induction on the unit interval $[0, 1]$ and some of its equivalents. Slides from a presentation, May 2010.

[93] W. Veldman and M. Bezem. Ramsey's theorem and the pigeonhole principle in intuitionistic mathematics. *Journal of the London Mathematical Society*, 2(47):193–211, 1993.

Index