TÉCNICO LISBOA

Bar recursion with abstract types

a study on the bounded functional interpretation

João Rafael Oliveira Mira da Silva

Thesis to obtain the Master of Science Degree in

Mathematics and Applications

Supervisors: Prof. Fernando Jorge Inocêncio Ferreira
Prof. João Filipe Quintas dos Santos Rasga

Examination Committee

Chairperson: Prof. Paulo Alexandre Carreira Mateus
Supervisor: Prof. João Filipe Quintas dos Santos Rasga
Members of the Committee: Prof. Gilda Maria Saraiva Dias Ferreira

October 2021

Logic may be unshakeable, but it cannot hold out
against a human being who wants to live.

— Franz Kafka, *The Trial*

Acknowledgments

I want to begin this text with some words of gratitude for those who, directly or indirectly, helped me bring this thesis to fruition.

Firstly, I want to thank my thesis advisor, Professor Fernando Ferreira. This whole adventure started with me — an unknown student from a different faculty — asking for research topics in the fascinating area of proof theory. I am deeply grateful to Prof. Fernando Ferreira for taking me in as a student, first as a supervisor for a research project, and now as an advisor for this thesis. I am thankful for the patient guidance, for the thoughtful suggestions, for the in-depth discussions, and for the lengthy meetings which never failed to renew my enthusiasm for logic.

I also want to thank Professor Cristina Sernadas, who introduced me to the world of logic and who was always available to talk about these topics and provide references for further study. It is with delight that I remember receiving many dropbox links with books whose depth I would only come to appreciate a few years later. When I was unsure about choosing my thesis advisor, it was Prof. Cristina Sernadas who encouraged me to apply for the research project with CMAFcIO.

Secondly, I want to thank my mom and grandma for always being there for me. I am indebted to them for their unconditional love and support. I am certain they have called me an uncountable number of times for dinner, and I know how often they were deprived of me while I was working on this document.

Thirdly, I want to thank my friends. I have a truly marvellous list of them, which this page is too small to contain. Even so, I want to write a few words of appreciation for some of the people dearest to me. To Carolina, for being my lighthouse when I thought I was lost. For all the Friday night dinners, in person or remotely, that were the only constant source of joy in these times of change. To Pedro, for being my lighthouse when I thought I was found. For all the time we spent geeking out about music, for all the toasts to Lucifer, and for reminding me to pick myself up and turn the world on its head.

Lastly, I want to thank CMAFcIO for their financial support during my research project and subsequent preparation of this thesis. I praise them for encouraging research among master students in mathematics, and I am glad for having been given a grant independently of my academic affiliations.

Resumo

As interpretações funcionais são uma ferramenta importante em teoria da demonstração. Estudamos duas extensões à interpretação funcional limitada de Ferreira e Oliva, no contexto da lógica clássica, que contribuem para a sua aplicabilidade em estudos de *proof mining.*

A primeira extensão introduz definições por bar recursão, uma forma de recursão transfinita em árvores bem fundadas. Com esta adição, a interpretação funcional limitada pode ser usada para analisar demonstrações que envolvam o princípio das escolhas dependentes e o princípio da compreensão numérica.

A segunda extensão diz respeito à inclusão de um tipo abstrato, isto é, à inclusão de um tipo base adicional. Este tipo pode representar um anel, ou um espaço métrico, por exemplo. Como consequência, a interpretação funcional limitada pode ser usada para a análise de demonstrações sobre estruturas matemáticas arbitrárias sem que estas tenham que ser codificadas.

O objetivo é estudar a extensão da interpretação funcional limitada simultaneamente com bar recursão e com um tipo abstrato. Verificamos em detalhe, na presença de um tipo abstrato, a dedução do princípio das escolhas dependentes a partir de bar indução, o princípio de demonstração associado à bar recursão. Numa aplicação, mostramos que a interpretação funcional limitada estendida completa automaticamente os espaços métricos.

Palavras-chave: teoria da demonstração, interpretação funcional, majorazibilidade, bar recursão, tipo abstrato, *proof mining*

Abstract

Functional interpretations are an important tool in proof theory. We study two extensions to the bounded functional interpretation of Ferreira and Oliva, in the context of classical logic, that contribute to its applicability in proof mining studies.

The first extension introduces definitions by bar recursion, a form of transfinite recursion on well-founded trees. With this addition, the bounded functional interpretation can be used to analyse proofs involving the principle of dependent choices and the principle of numerical comprehension.

The second extension concerns the addition of an abstract type, that is, the inclusion of an additional ground type. This type can represent a ring, or a metric space, for example. As a consequence, the bounded functional interpretation can be used for the analysis of proofs about arbitrary mathematical structures without these having to be encoded.

Our goal is to study the simultaneous extension of the bounded functional interpretation with bar recursion and an abstract type. We check in detail, in the presence of an abstract type, the deduction of the principle of dependent choices from bar induction, the proof principle corresponding to bar recursion. As an application, we show that the extended bounded functional interpretation automatically completes metric spaces.

Keywords: proof theory, functional interpretation, majorisability, bar recursion, abstract type, proof mining

Contents

Chapter 1

Introduction

A *proof interpretation* of a formal system A in a formal system B consists of a mapping of the formulas of A into formulas of B. In addition, we require that there is a computable translation of the proofs in A to proofs in B in such a way that the theorems of A are mapped by the interpretation to theorems of B. This translation is usually called the soundness theorem of the interpretation.

An important proof interpretation was defined by Kurt Gödel in [21], and is known as the *dialectica* interpretation, after the name of the journal where it was published. The dialectica interpretation interprets Heyting arithmetic HA — the intuitionistic version of Peano arithmetic — in a quantifier-free system T, known as the system of primitive recursive functionals of finite-type. Combined with a negative translation of classical logic to intuitionistic logic, Gödel's interpretation can be used to provide a syntactic proof of the consistency of Peano arithmetic. An extension of the dialectica interpretation with *bar recursive* functionals, carried out by Clifford Spector in [39], strengthens this result to a relative consistency proof of full second-order arithmetic. We can also say this is a proof of the relative consistency of analysis, since many results from this field can be stated and proven in second-order arithmetic.

Nowadays, the more interesting facet of the dialectica interpretation is that it can be used to provide terms that are witnesses for existential theorems, given that these are proven by restricted means. This term extraction is now a core part of the *proof mining* programme, originally known as *proof unwinding*, that stems from the work of Georg Kreisel [29, 30, 31], being further developed by Ulrich Kohlenbach. The main question of this programme is posed by Kreisel:

> What more do we know if we have proved a theorem by restricted means
> than if we merely know that it is true?

Currently, a considerable part of the programme is concerned with extracting new effective information from proofs that use some non-constructive principles. Typical results in proof mining are computable uniform bounds for theorems in analysis, of which many examples can be seen in [28, §15–18]. The main technique at work here is the *monotone functional interpretation* of Kohlenbach [26] that combines the dialectica interpretation with the hereditary majorisability relation of William Howard [23]. Unlike the dialectica interpretation, that provides exact witnessing terms, the monotone interpretation allows only for the extraction of bounds for these terms — in this context called *majorants*. However, in exchange for the weaker term extraction, the monotone interpretation can deal with non-constructive principles that cannot be dealt with using the dialectica interpretation directly, including, for instance, weak König's lemma — the claim that every infinite binary tree has an infinite path. The aforementioned interpretations rely on typed formal systems where the terms are assigned a finite type, that is, either the type of the natural numbers, or the type of a function between finite types. An extension of the monotone functional interpretation includes an additional *abstract type* that represents an abstract space, for instance, a metric space, or a normed space, see [27, 20]. The abstract type lifts the restriction of working only with spaces that have to be encoded.

A different interpretation used for proof mining is the *bounded functional interpretation* of Fernando Ferreira and Paulo Oliva [19], a functional interpretation that uses the strong majorisability relation of Marc Bezem [3]. Like the monotone interpretation, the bounded functional interpretation focuses on the extraction of bounding terms, instead of precise witnessing terms, and, due to this, it can be used for the analysis of proofs involving some non-constructive principles. The main difference is that the former interpretation only takes majorisability into account after the interpretation of formulas, while the latter has the majorisability relation built into the language of its formal system. The interpretation of weak König's lemma is performed using a central principle of the bounded functional interpretation — the principle of bounded collection bC — that is essentially a generalisation of weak König's lemma to higher types. This principle clashes with set-theoretic truth, and it is responsible for a phenomenon that we may call injection of uniformities, or introduction of ideal elements: it can be used to prove the existence of an object satisfying a global property, given the existence of objects satisfying local properties. For example, this principle can be used to prove that every Cauchy sequence converges, provided that it has a computable rate of convergence.

In this work, we present a version of the bounded functional interpretation directly for classical logic [12], inspired by Joseph Shoenfield's direct interpretation of Peano arithmetic in [37, §8.3]. We follow two lines of inquiry, proposed by Ferreira in [13]

- an extension with bar recursive functionals,
- an extension with an abstract type.

These extensions to the bounded functional interpretation have been studied individually in the PhD thesis of Patrícia Engrácia [9], in the context of intuitionistic logic. Our contributions are twofold,

- the simultaneous extension with bar recursive functionals and an abstract type,
- the study of these extensions directly for classical logic, instead of through a negative translation.

This former item settles a question posed by Engrácia and Ferreira in [11], and is relevant for the proof mining programme. On one hand, the extension with bar recursion enables the analysis of proofs using stronger non-constructive principles, such as the principle of dependent choices and the principle of full numerical comprehension. On the other hand, the extension with an abstract type allows proof mining to go beyond representable spaces and to consider theorems about arbitrary spaces. Needless to say that the coexistence of both extensions is of practical importance to the proof mining practice in the context of the bounded functional interpretation.

Thesis outline

In Chapter 2, we present the bounded functional interpretation. We begin by describing the theory of arithmetic in all finite types — the base theory for the formal systems interpreted by the bounded functional interpretation — then we define some of its models, and present the strong majorisability relation. After this, we present the intensional version of strong majorisability, governed by a rule, and how this version relates to its extensional counterpart. Having developed the required background, we define the bounded functional interpretation for classical logic, presenting also its characteristic principles and the soundness and characterisation theorems.

In Chapter 3, we describe the extension of the bounded functional interpretation with bar recursive functionals. First, we describe the principle of definition by bar recursion in an informal setting, before stating the bar recursor axioms. Following this, we check that the model of strongly majorisable functionals is a model for the bar recursor axioms. Then, we present the principle of proof by bar induction — the principle of proof corresponding to bar recursion — first presenting this principle informally and then proving that it holds for an extension of the theory of arithmetic in all finite types. With bar induction, we are afterwards able to prove the principle of dependent choices and the principle of full numerical comprehension. As an application, we show how the bar recursive functionals enable the interpretation of second-order arithmetic. Furthermore, we check that having only the simplest bar recursor is sufficient for the interpretation of the principle of arithmetical comprehension.

In Chapter 4, we consider the extension of the bounded functional interpretation simultaneously with bar recursive functionals and with an abstract type. We begin by extending the theory of finite-type arithmetic with an abstract type, and by extending the strong majorisability relation to encompass the new types. After this, we illustrate the general concept of an abstract type with some concrete examples. Having laid the necessary foundations, we define the bounded functional interpretation in a version extended to abstract types, presenting then its characteristic principles and main theorems. Following the steps of Chapter 3, we augment the interpretation with bar recursion and bar induction. In a similar way, we prove the principle of dependent choices and numerical comprehension, now in the presence of an abstract type. We conclude with an application showing that metric spaces are automatically completed by the bounded functional interpretation.

Chapter 2

Bounded functional interpretation

An *interpretation* of a formal system A in a formal system B is a function mapping the formulas of A to formulas of B in an effective manner. This function is defined in a way that relates the theorems of A with the theorems of B. The name *functional* arises from the existence of functionals — functions that take other functions as arguments — in our theories of interest. In fact, a functional interpretation effects a trade-off between quantifiers and higher-order functionals.

In this chapter, we describe the *bounded functional interpretation* — an interpretation of the theory of arithmetic in all finite types.

- In Section 2.1, we present the theory of finite-type arithmetic, first describing the finite types, and then the language and axioms of this theory. We define the standard set-theoretic model and the model of strongly majorisable functionals based on the relation of strong majorisability.
- In Section 2.2, we consider the theory of finite-type arithmetic with intensional majorisability, a version of strong majorisability partially governed by a rule, and how this relates to extensional strong majorisability.
- In Section 2.3, we define the bounded functional interpretation and its characteristic principles, giving several examples of applications of the principle of bounded collection. We also state the soundness and characterisation theorems.

2.1 Arithmetic in all finite types

The theory of *arithmetic in all finite types* is a many-sorted theory, with a sort for each finite type.

Types The *finite types* are defined recursively as follows:

- 0 is a finite type — the *ground type*,
- if σ and τ are finite types, then $\sigma \to \tau$ is a finite type — a *higher type*.

Intuitively, the ground type 0 represents the type of the natural numbers, and the type $\sigma \to \tau$ stands for the type of total functions from objects of type σ to objects of type τ. We assume that $\to$ associates to the right, that is, the notation

$$\tau_1 \to \tau_2 \to \cdots \to \tau_n \quad \text{stands for} \quad \tau_1 \to (\tau_2 \to (\cdots \to \tau_n)\dots).$$

Objects of a type $(\sigma \to \tau) \to \rho$ have arguments of type $\sigma \to \tau$ and are usually called *functionals*.

Language The *language* of finite-type arithmetic has

- variables $x^\tau, y^\tau, z^\tau, \dots$ of each type τ,
- logical constants
  - $\Pi_{\sigma,\tau}$ of type $\sigma \to \tau \to \sigma$, for all types σ and τ,
  - $\Sigma_{\rho,\sigma,\tau}$ of type $(\rho \to \sigma \to \tau) \to (\rho \to \sigma) \to \rho \to \tau$, for all types ρ, σ and τ,
- arithmetical constants
  - 0 of type 0,
  - S of type $0 \to 0$,
  - R_τ of type $0 \to \tau \to (\tau \to 0 \to \tau) \to \tau$, for all non-empty tuples of types τ
- a binary predicate symbol $=_0$ between objects of type 0.

The logical constants Π and Σ are called the *combinators* and they correspond to the K and S combinators from SKI combinatory logic, see [8]. The combinator Π represents the function $(x, y) \mapsto x$ and Σ represents the function $(x, y, z) \mapsto x(z, y(z))$.

The arithmetical constant 0 represents the first natural number and S represents the successor function $n \mapsto n+1$. The constants R_τ are called the *recursors* and are used to define functions by recursion, they represent the function g such that $g(0, x, f) = x$, and $g(n+1, x, f) = f(g(n, x, f), n)$. If τ is a tuple with more than one element, then R_τ is itself a tuple with the same number of elements called *simultaneous recursor*. To simplify the notation, when denoting (possibly empty) tuples of types, variables or terms, we will not employ any additional symbols — we might write x for the tuple $(x_1, \dots, x_n)$.

There is only a predicate symbol for equality between objects of type 0, and not for equality between objects of higher types. This minimal treatment of equality follows [42] and is due to the fact that functional interpretations are sensitive to the way equality is treated. Indeed, we will see ahead that the theory of finite-type arithmetic with some added axioms is not compatible with substitution by pointwise equal functionals for every type — in fact it refutes the axiom of extensionality (see Example 2.8).

Terms The *terms* of the language of finite-type arithmetic are typed — they are assigned a type. The terms of type τ are:

- constants of type τ,
- variables of type τ,
- terms obtained by functional application — if t is a term of type $\sigma \to \tau$ and s is a term of type σ, then the term t applied to s, denoted by ts, is a term of type τ.

The variables of a term are the variables that occur in that term. When a term has no variables, we say that the term is *closed*. We assume that application associates to the left, i.e., the notation

$$ts_1 s_2 \dots s_n \quad \text{stands for} \quad (\dots((ts_1)s_2)\dots)s_n.$$

To improve readability, we will often write this as $t(s_1, s_2, \dots, s_n)$ instead. When referring a variable x that might occur in a term, we write $t[x]$ instead of $t(x)$ to avoid confusion with functional application.

Terms are usually written with the type superscript notation t^τ — meaning that t is a term of type τ — similar to the type superscripts used for variables. When there is no potential ambiguity, we will omit the type superscript for both variables and terms.

Formulas The *atomic formulas* are of the form $t =_0 s$ with t and s terms of type 0. The *formulas* are then defined recursively:

- atomic formulas are formulas,
- if A is a formula, then $\neg A$ is a formula,
- if A and B are formulas, then $A \vee B$ is a formula;
- if A is a formula, then $\forall x^\tau A$ is a formula, for any variable x^τ of type τ.

Since we will be using classical logic (as opposed to intuitionistic logic), the other connectives and the existential quantifier are taken as abbreviations:

- $A \wedge B$ is $\neg(\neg A \vee \neg B)$,
- $A \to B$ is $\neg A \vee B$,
- $\exists x^\tau A$ is $\neg \forall x^\tau \neg A$.

The primitive connectives are chosen to be $\neg$, $\vee$ and $\forall$ because the version of the bounded functional interpretation relevant for this work is based on the interpretation by Shoenfield in [37, §8.3], which in turn uses the proof calculus in [37, §2.6] with these primitive connectives. (Any sound and complete proof calculus could be used in place of this one.)

When writing formulas, some parenthesis can be unambiguously omitted and the connective precedence is taken as (from higher to lower)

1. $\neg$, $\forall$ and $\exists$,
2. $\wedge$ and $\vee$,
3. $\to$ and $\leftrightarrow$.

Formulas without the quantifiers $\forall$ and $\exists$ are said to be *quantifier-free*. A formula of the form $\forall x\, A$, with A quantifier-free, is said to be *universal*. Likewise, $\exists x\, A$ formulas are said to be *existential*. In these schemata, the variable x can be a tuple of variables $(x_1, \dots, x_n)$ and in that case the quantifier $\forall x$ should be read as multiple quantifiers $\forall x_1 \dots \forall x_n$, and the same applies for $\exists x$.

The *free* variables of a formula are defined recursively:

- if A is an atomic formula, all variables occurring in A are free,
- the free variables of $\neg A$ are the free variables of A,

- the free variables of $A \vee B$ are the free variables of A and the free variables of B,
- the free variables of $\forall x\, A$ are the free variables of A other than x.

When a variable occurs in a formula but it is not free, we say it is *bound*. Formulas with no free variables are said to be *sentences*. To refer to a free variable x of a formula A we write $A(x)$, however, we also use the same notation if x does not occur in A — the notation $A(x)$ means that if x occurs in A, then it must occur free.

The notation we use for substitution is $A[t/x]$ and it represents the formula A with the variable x being replaced by the term t whenever x occurs free — i.e., not in the scope of a quantifier over x — oftentimes we use the shorter notation $A(t)$. We say that t is *free* for x in A when no variables in t become bound in $A[t/x]$. To avoid substitutions $A[t/x]$ with t not free for x in A, we assume that the bound variables of A are renamed so that there are no clashes with the variables of t.

Axioms The *theory of Peano arithmetic in all finite types*, denoted by PA^ω (from Peano arithmetic), is based on classical logic together with *axioms* that are the universal closures of the following formulas:

- equality axioms

  $x =_0 x$

  $x =_0 y \wedge A[x/w] \to A[y/w]$

  where $A(w)$ is an atomic formula,
- successor axioms

  $Sx \neq_0 0$

  $Sx =_0 Sy \to x =_0 y$,
- combinator axioms

  $A[\Pi xy/w] \leftrightarrow A[x/w]$

  $A[\Sigma xyz/w] \leftrightarrow A[xz(yz)/w]$

  where $A(w)$ is an atomic formula,
- recursor axioms

  $A[R(0, y, z)/w] \leftrightarrow A[y/w]$

  $A[R(Sx, y, z)/w] \leftrightarrow A[z(R(x, y, z), x)/w]$

  where $A(w)$ is an atomic formula,
- induction schema

  $A(0) \wedge \forall n^0\, (A(n) \to A(Sn)) \to \forall n^0\, A(n)$

  where $A(n)$ is an arbitrary formula.

Using these same axioms with intuitionistic logic, we get the theory of Heyting arithmetic in all finite types and denoted by HA^ω — the intuitionistic correspondent of PA^ω.

The equality axioms can be used to prove symmetry and transitivity for $=_0$. Moreover, the equality, combinator and recursor axioms that are restricted to atomic formulas $A(w)$ can be proven to hold for arbitrary formulas $A(w)$.

In addition to the primitive equality $=_0$ between objects of type 0, we define equality for higher types as pointwise or extensional equality, that is, given terms t and s of type $\sigma \to \tau$, we define

$$t =_{\sigma\to\tau} s \quad \text{as} \quad \forall x^\sigma \, (tx =_\tau sx),$$

assuming that x is not a variable of t or of s. When it is clear from context, we omit the type subscript for equality — instead of writing $=_{\sigma\to\tau}$ we write only $=$.

Some of the axioms above are equalities in disguise, e.g. the axiom for the combinator Π given by $A[\Pi xy/w] \leftrightarrow A[x/w]$. With equality $=_\tau$ defined for every type, it seems that we opted for an inelegant axiom instead of $\Pi xy = x$. This convoluted way of writing equalities is due to our issues with the axiom of extensionality

$$x = y \wedge A[x/w] \to A[y/w],$$

where $A(w)$ is an atomic formula. This axiom can be derived from the instance $x = y \to zx = zy$, when the language only has the relation symbol $=_0$. In Example 2.8, we see that PA^ω with some additional axioms proves the negation of the axiom of extensionality.

An interesting alternative is to have equality — logical equality — for each type, instead of having only primitive equality for type 0. By logical equality, we mean that equality would be governed by axioms similar to the equality axioms above, instead of being defined as extensional or pointwise equality. This variant is known in the literature as the neutral theory $\mathsf{N}\text{-}\mathsf{PA}^\omega$, see [41, §1.6.3–§1.6.7] and [2]. The reason we do not use this variant is historical: the first functional interpretation, Gödel's dialectica interpretation [21], requires that quantifier-free formulas $A(x)$ have a characteristic term $t(x)$, in the sense that $A(x) \leftrightarrow t(x) = 0$. Such terms can be proven to exist in PA^ω, but not in $\mathsf{N}\text{-}\mathsf{PA}^\omega$, where there is no characteristic term for the formula $f =_{0\to 0} g$, for instance. The bounded functional interpretation, inspired by the dialectica interpretation, was thus introduced in [19] using only $=_0$ as primitive, even though the existence of characteristic terms seems to be no longer necessary for the former interpretation.

Combinatorial completeness The combinators provide the theory PA^ω with *combinatorial completeness* — for each term t and variable x, we can construct a term $\lambda x.t$ whose variables are those of t other than x and such that PA^ω proves

$$A[(\lambda x.t)s/w] \leftrightarrow A[t[s/x]/w],$$

for any formula A and term s.

With the recursor R_0, we can construct a closed term for each primitive recursive function. Therefore, the usual first-order Peano arithmetic PA can be seen as a subsystem of PA^ω. Moreover, using the recursors R_τ for higher types, we can construct closed terms in PA^ω for functions that are not primitive recursive, e.g. the Ackermann function. Since the theory PA^ω is based on the theory T introduced by Gödel in [21], the functionals defined by closed terms of PA^ω are called *Gödel primitive recursive functionals of finite type*. At the type $0 \to 0$, the Gödel primitive recursive functionals coincide with the provably recursive functions of PA — the functions f for which there is a formula $A(x, y)$ such that $f(x) = y$ if and only if $A(x, y)$ and PA proves $\forall x \, \exists! y \, A(x, y)$, where $\exists! y$ abbreviates existence of a unique y — see [28, §3.3] for more details.

Order The usual order relation between objects of type 0 can be defined using primitive recursive functions:

- the predecessor function P, satisfying $P(0) = 0$ and $P(Sx) = x$,
- and cut-off subtraction $\dot{-}$, satisfying $x \dot{-} 0 = x$ and $x \dot{-} Sy = P(x \dot{-} y)$.

We then define $x \leq_0 y$ to be $x \dot{-} y = 0$, and, for higher types, we define $\leq_{\sigma \to \tau}$ as pointwise inequality — in other words, given terms t and s of type $\sigma \to \tau$, we define

$$t \leq_{\sigma \to \tau} s \quad \text{as} \quad \forall x^\sigma \, (tx \leq_\tau sx),$$

assuming that x is not a variable of t or of s. The maximum operator can also be defined: for the base type we define $\max_0$ as $\lambda x^0, y^0.[x + (y \dot{-} x)]$, and, for higher types, we define $\max_{\sigma \to \tau}$ as pointwise maximum, i.e., we write

$$\max_{\sigma \to \tau} \quad \text{to mean} \quad \lambda t^{\sigma \to \tau}, s^{\sigma \to \tau}, x^\sigma . \max_\tau(tx, sx).$$

As with equality, we remove the type subscripts and write only $\leq$ and max when the types can be inferred from context. The maximum of a finite set of objects can be defined using the binary maximum operator — given a term t of type $0 \to \tau$, we define the term $\max\{ti \mid i \leq \cdot\}$ recursively by

- $\max\{ti \mid i \leq 0\}$ is $t(0)$,
- $\max\{ti \mid i \leq Sx\}$ is $\max(t(Sx), \max\{ti \mid i \leq x\})$.

Standard model The *full set-theoretic model*, denoted by $\mathscr{S}^\omega$, is the standard model for finite-type arithmetic. The type structure $\mathscr{S}^\omega$ is defined recursively as

- $S_0 = \mathbb{N}$,
- $S_{\sigma \to \tau} = S_\tau^{S_\sigma}$, for all types σ and τ.

Over this type structure, we consider natural interpretations for the function symbols of the language of finite arithmetic — 0 is interpreted as $0 \in \mathbb{N}$, the symbol S is interpreted as the successor function $n \mapsto n + 1$, et cetera. Not only is $\mathscr{S}^\omega$ a model of PA^ω, it is also a model of the axiom of extensionality.

Majorisability The *strong majorisability* relation defined by Bezem in [3] is a variation of the hereditary majorisability relation of Howard [23]. We define the strong majorisability relation, written as $\leq^*_\tau$, and define sets M_τ simultaneously by recursion on the type τ:

- $n \leq^*_0 m$ is simply $n \leq m$ and we define $M_0 = \mathbb{N}$,
- $x \leq^*_{\sigma \to \tau} y$ holds when $x, y \in M_\tau^{M_\sigma}$ and

$$\forall u, v \in M_\sigma \, (u \leq^*_\sigma v \to xu \leq^*_\tau yv \wedge yu \leq^*_\tau yv),$$

  and we define $M_{\sigma \to \tau} = \{x \in M_\tau^{M_\sigma} \mid x \leq^* x^*, \text{ for some } x^* \in M_\tau^{M_\sigma}\}$, for all types σ and τ.

Intuitively, having $x \leq^* y$ means that y is an upper bound for x, and that y is monotone. Indeed, if $x \leq^* x$, then we say that x is *monotone* with respect to the relation $\leq^*$, in the sense that if $u \leq^* v$, then $xu \leq^* xv$. This entails that $\leq^*$ is not reflexive, since there are objects that are not monotone — for example, the function $f : \mathbb{N} \to \mathbb{N}$ such that

$$f(n) = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{if } n > 0, \end{cases}$$

is not monotone, hence it is not the case that $f \leq^* f$. However, there are some objects for which $\leq^*$ is reflexive: if we have $x \leq^* y$, then $y \leq^* y$. Furthermore, the relation $\leq^*$ is transitive: when we have $x \leq^* y$ and $y \leq^* z$, it follows that $x \leq^* z$.

The model of *strongly majorisable functionals*, denoted by $\mathscr{M}^\omega$, contains functionals that can be strongly majorised, the type structure for $\mathscr{M}^\omega$ being defined with the sets M_τ defined above. This type structure is closed under function application — if $f \in M_{\sigma\to\tau}$ and $x \in M_\sigma$, then $f(x) \in M_\tau$.

With natural interpretations for the function symbols of the language of PA^ω, it is possible to find strong majorants for these function symbols, and hence we conclude that $\mathscr{M}^\omega$ is a model of PA^ω (it is also a model of the axiom of extensionality). For a proof, see [28, Proposition 3.69].

The models $\mathscr{S}^\omega$ and $\mathscr{M}^\omega$ are different — there are functionals that cannot be strongly majorised. For type $0 \to 0$, we still have equality $M_{0\to 0} = S_{0\to 0} = \mathbb{N}^\mathbb{N}$, since any function $f : \mathbb{N} \to \mathbb{N}$ is strongly majorised by $f^M(n) = \max_{k<n} f(k)$. However, for type $(0 \to 0) \to 0$, we have $M_{(0\to 0)\to 0} \neq S_{(0\to 0)\to 0}$. In fact, consider the following example from [13, §2.4]: let $\Sigma : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ be such that

$$\Sigma(f) = \begin{cases} n & \text{if } n \text{ is the least } k \text{ such that } f(k) \neq 0 \\ 0 & \text{if } f(k) = 0, \text{ for all } k. \end{cases}$$

We claim that there is no $\Psi : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ such that $\Sigma \leq^* \Psi$. If this was the case, then, in particular, for every $f : \mathbb{N} \to \mathbb{N}$ such that $f \leq^* 1^{0\to 0}$, we would have $\Sigma(f) \leq \Psi(1^{0\to 0})$, where $1^{0\to 0}$ is the constant function with value 1 — this is, if $f(k) \leq 1$ for every k, then the least k for which $f(k) \neq 0$ would be at most $\Psi(1^{0\to 0})$. This is a contradiction: define a function f such that

$$f(k) = \begin{cases} 1 & \text{if } k = \Psi(1^{0\to 0}) + 1 \\ 0 & \text{otherwise.} \end{cases}$$

Then $f \leq^* 1^{0\to 0}$, but $\Sigma(f) = \Psi(1^{0\to 0}) + 1 \not\leq \Psi(1^{0\to 0})$. Thus $\Psi \notin M_{(0\to 0)\to 0}$.

Other models of PA^ω include the term model, see [34] and [40], and the model of sequentially continuous functionals, see [36].

2.2 Intensional majorisability

The strong majorisability relation is a key component of the bounded functional interpretation. The definition in the previous section can be used to define strong majorisability within finite-type arithmetic — that is, we define recursively the symbols $\leq^*$ within PA^ω as

- $t \leq^*_0 s$ is defined as $t \leq_0 s$,

- $t \leq^*_{\sigma\to\tau} s$ is defined as

$$\forall u^\sigma, v^\sigma\, (u \leq^*_\sigma v \to tu \leq^*_\tau sv \wedge su \leq^*_\tau sv),$$

  for all types σ and τ, assuming that u and v are not variables of t or of s.

This definition we call *extensional* strong majorisability.

There is a different definition — of *intensional* strong majorisability — which is relevant for the bounded functional interpretation. The intensional approach consists of adding an inference rule for strong majorisability which is used to tackle the issue of quantifier complexity. In the next section, it will become clear that the bounded functional interpretation is more compatible with universal axioms (see Theorem 2.11). Extensional strong majorisability is governed by axioms of the form

$$t \leq^* s \leftrightarrow \forall u, v\, (u \leq^* v \to tu \leq^* sv \wedge su \leq^* sv).$$

The problem is that these axioms are not universal. In particular, the axiom

$$\forall u, v\, (u \leq^* v \to tu \leq^* sv \wedge su \leq^* sv) \to t \leq^* s$$

is not universal, since its prenex form is

$$\exists u, v\, \big((u \leq^* v \to tu \leq^* sv \wedge su \leq^* sv) \to t \leq^* s\big).$$

Apropos, this is also the reason why extensional equality is problematic in our setting: we would have to include an axiom of the form $\forall x\, (tx = sx) \to t = s$, which has prenex form $\exists x\, (tx = sx \to t = s)$.

The *intensional* version of strong majorisability involves adding to the language of finite-type arithmetic, for each type τ, a binary relation symbol $\trianglelefteq_\tau$ between objects of type τ. The symbol $\trianglelefteq_\tau$ is meant to represent the strong majorisability relation $\leq^*_\tau$. As before, we omit the type subscript and write only $\trianglelefteq$ whenever no ambiguity arises.

The atomic formulas then include also formulas of the form $t \trianglelefteq_\tau s$. In addition to the usual quantifiers, there are also *bounded quantifiers*:

- if A is a formula, and the variable x does not occur in the term t, then $\forall x \trianglelefteq_\tau t\, A$ is a formula.

As before, we define the bounded existential quantifier as an abbreviation:

- $\exists x \trianglelefteq_\tau t\, A$ is $\neg\forall x \trianglelefteq_\tau t\, \neg A$.

A formula whose quantifiers are all bounded is said to be a *bounded* formula. If A_{bd} is a bounded formula, then formulas of the form $\forall x\, A_{bd}$ are said to be *universal bounded* and formulas of the form $\exists x\, A_{bd}$ are said to be *existential bounded.* In these definitions, we allow x to be a tuple of variables.

The *theory of Peano arithmetic in all finite types with intensional majorisability*, denoted by $\mathsf{PA}^\omega_\trianglelefteq$, is based on PA^ω with its axioms now applying to the formulas of the extended language, in addition to the universal closures of:

- bounded quantifier axioms
  $\forall x \trianglelefteq_\tau t\, A \leftrightarrow \forall x\, (x \trianglelefteq_\tau t \to A)$

$[\exists x \trianglelefteq_\tau t\ A \leftrightarrow \exists x\,(x \trianglelefteq_\tau t \wedge A)$, a consequence of the abbreviation $\exists x \trianglelefteq_\tau t\ A]$

where A is an arbitrary formula, the variable x does not occur in the term t, and τ is any type,

- strong majorisability axioms

  $x \trianglelefteq_0 y \leftrightarrow x \leq_0 y$

  $x \trianglelefteq_{\sigma\to\tau} y \to \forall u^\sigma, v^\sigma\,(u \trianglelefteq_\sigma v \to xu \trianglelefteq_\tau yv \wedge yu \trianglelefteq_\tau yv)$

  for all types σ and τ,

and the strong majorisability rule

$$\frac{A_{bd} \wedge u \trianglelefteq_\sigma v \to tu \trianglelefteq_\tau sv \wedge su \trianglelefteq_\tau sv}{A_{bd} \to t \trianglelefteq_{\sigma\to\tau} s}$$

where A_{bd} is a bounded formula, and u and v do not occur free in A_{bd}, in t or in s.

The strong majorisability rule is the fundamental difference between extensional and intensional majorisability. This rule avoids having to introduce the extensional axiom that was not universal. In essence, it deactivates the computational content of the strong majorisability relation. At the same time, the intensional theory $\mathsf{PA}^\omega_\trianglelefteq$ does not enjoy a deduction theorem precisely because of the added rule, see [19, Proposition 8].

The intensional theory $\mathsf{PA}^\omega_\trianglelefteq$ can prove the aforementioned properties of strong majorisability:

- $x \trianglelefteq y \to y \trianglelefteq y$,
- $x \trianglelefteq y \wedge y \trianglelefteq z \to x \trianglelefteq z$.

For a proof, see [19, Lemma 2].

Flattening The process of translating formulas of the intensional theory $\mathsf{PA}^\omega_\trianglelefteq$ into formulas of the extensional theory PA^ω with the defined strong majorisability relation $\leq^*$ is called *flattening*. Given a formula A of $\mathsf{PA}^\omega_\trianglelefteq$, its flattening, denoted by A^*, is the formula of PA^ω obtained from A by replacing the relation symbols $\trianglelefteq_\tau$ with $\leq^*_\tau$, and the occurrences of bounded quantifiers $\forall x \trianglelefteq_\tau t\ A$ with $\forall x\,(x \leq^*_\tau t \to A)$.

If the theory $\mathsf{PA}^\omega_\trianglelefteq$ proves A, then the theory PA^ω proves its flattening A^*. Indeed, all applications of the rule of strong majorisability can be replaced using the defining axiom

$$t \leq^* s \leftrightarrow \forall u, v\,(u \leq^* v \to tu \leq^* sv \wedge su \leq^* sv).$$

For example, by flattening it follows that the theory PA^ω proves

- $x \leq^* y \to y \leq^* y$,
- $x \leq^* y \wedge y \leq^* z \to x \leq^* z$,

since the intensional theory proves the corresponding theorems for $\trianglelefteq$.

2.3 Bounded functional interpretation

The original functional interpretation — Gödel's dialectica interpretation [21] — interpreted Peano arithmetic in two steps: first a negative translation of classical Peano arithmetic into intuitionistic Heyting arithmetic, and then the interpretation itself mapped Heyting arithmetic into theory T, which is essentially the quantifier-free fragment of PA^ω. A posteriori, Shoenfield defines an interpretation of Peano arithmetic with a single step in [37, §8.3]. Both interpretations hinge on a mapping of formulas into a $\forall\exists$-form and on providing precise witnessing terms for existential theorems. The aim of Gödel with this interpretation was to prove the consistency of Peano arithmetic via a syntactic argument, relative to the consistency of the quantifier-free theory T.

The bounded functional interpretation of Ferreira and Oliva provides an interpretation of finite-type Peano arithmetic (with intensional majorisability) into itself. When it was first introduced in [19], the bounded functional interpretation of $\mathsf{PA}^\omega_\trianglelefteq$ was also defined in two steps: first using a negative translation by Kuroda [33], followed by the interpretation itself. Later, Ferreira [12] defines an interpretation in the style of Shoenfield, with a single step. Again, both interpretations are based on a mapping of formulas into a $\forall\exists$-form, but now the focus is on obtaining bounds for witnesses of existential theorems in lieu of the witnesses themselves.

Using a majorisability relation enables the shift of focus from precise witnesses to bounds for witnesses. This seemingly innocuous change broadens the applicability of functional interpretations to proofs using stronger proof-theoretical principles. It can be the case that non-computable objects, whose existence is proven using ineffective principles, can be bounded by a computable object, as long as the existence claim has the right logical form — essentially the form $\forall x\, \exists y\, A_{bd}(x, y)$, where $A_{bd}(x, y)$ is a bounded formula. The example used in [19] is weak König's lemma — the claim that every infinite binary tree has an infinite path — this is ineffective since there are (infinite) recursive binary trees with no infinite recursive path. An example of a tree with this property is known as the Kleene binary tree, introduced in [25]. The infinite path is given by its characteristic function, and the bound is simply the constant function with value 1. In the context of reverse mathematics, weak König's lemma is equivalent, in terms of proof-theoretic strength, to well-known ineffective principles from analysis such as

- the Heine-Borel covering lemma: every covering of the closed interval $[0, 1]$ by a sequence of open intervals has a finite subcovering,
- the maximum principle: every continuous real-valued function on $[0, 1]$ attains a (finite) supremum.

Other principles with the same strength as weak König's lemma are listed in [38, Theorem I.10.3]. It can also be seen that weak König's lemma is an instance of the contrapositive of the principle of bounded collection bC, a characteristic principle of the bounded functional interpretation (see Example 2.3).

In this section, we present the bounded functional interpretation directly for classical logic, employing the Shoenfield-like interpretation presented in [12].

An object x is monotone whenever x strongly majorises itself. We define an abbreviation for quantifiers ranging over monotone objects, called *monotone quantifiers*:

- $\widetilde{\forall} x\, A$ is an abbreviation for $\forall x\, (x \trianglelefteq x \to A)$,
- $\widetilde{\exists} x\, A$ is an abbreviation for $\exists x\, (x \trianglelefteq x \wedge A)$,
- $\widetilde{\forall} x \trianglelefteq t\ A$ is an abbreviation for $\forall x \trianglelefteq t\, (x \trianglelefteq x \to A)$,
- $\widetilde{\exists} x \trianglelefteq t\ A$ is an abbreviation for $\exists x \trianglelefteq t\, (x \trianglelefteq x \wedge A)$.

Monotone quantifiers $\widetilde{\forall} x\, A$ are not bounded quantifiers $\forall x \trianglelefteq t\ A$ — in a bounded quantifier, t is a term in which the variable x does not occur.

Definition 2.1 (Bounded functional interpretation)**.** To each formula A of the language of $\mathsf{PA}^{\omega}_{\trianglelefteq}$ we assign a formula A^U, called the *bounded functional interpretation* of A, which is of the form

$$\widetilde{\forall} x\ \widetilde{\exists} y\ A_U(x, y),$$

with $A_U(x, y)$ a bounded formula, and x and y are (possibly empty) tuples of variables. The formulas A^U and A_U are defined recursively as follows:

- if A is an atomic formula, then A^U and A_U are simply A.

For the remaining cases, let the interpretation of A be $\widetilde{\forall} x\, \widetilde{\exists} y\ A_U(x, y)$ and the interpretation of B be $\widetilde{\forall} u\, \widetilde{\exists} v\ B_U(u, v)$. Then:

- $(\neg A)^U$ is $\widetilde{\forall} f\ \widetilde{\exists} x\ \widetilde{\exists} x' \trianglelefteq x\ \neg A_U(x', fx')$,
- $(A \vee B)^U$ is $\widetilde{\forall} x, u\ \widetilde{\exists} y, v\ (A_U(x, y) \vee B_U(u, v))$,
- $(\forall z\, A)^U$ is $\widetilde{\forall} w, x\ \widetilde{\exists} y\ \forall z \trianglelefteq w\ A_U(x, y)$,
- $(\forall z \trianglelefteq t\ A)^U$ is $\widetilde{\forall} x\ \widetilde{\exists} y\ \forall z \trianglelefteq t\ A_U(x, y)$.

The formula inside the unbounded $\widetilde{\forall}\, \widetilde{\exists}$ quantifiers of A^U is defined to be A_U, that is:

- $(\neg A)_U$ is $\widetilde{\exists} x' \trianglelefteq x\ \neg A_U(x', fx')$,
- $(A \vee B)_U$ is $A_U(x, y) \vee B_U(u, v)$,
- $(\forall z\, A)_U$ is $\forall z \trianglelefteq w\ A_U(x, y)$,
- $(\forall z \trianglelefteq t\ A)_U$ is $\forall z \trianglelefteq t\ A_U(x, y)$.

The interpretations for the defined connectives $\wedge$, $\to$, and $\exists$ can be computed using their abbreviations.

Bounded formulas are left unchanged by the interpretation — if A is a bounded formula, then both A^U and A_U coincide with A. The clause for negation is the only one responsible for the increase in type complexity. Indeed, when A^U is $\widetilde{\forall} x^{\sigma}\ \widetilde{\exists} y^{\tau}\ A_U(x, y)$, then $(\neg A)^U$ has a universally quantified variable f of type $\sigma \to \tau$. The motivation behind the definition of $(\neg A)^U$ pertains to the axiom of choice, which can be written in the form

$$\forall x\, \exists y\ A(x, y) \leftrightarrow \exists f\, \forall x\ A(x, fx),$$

where the implication from left to right is what is usually called the axiom of choice. Taking $A(x, y)$ as $A_U(x, y)$, the negative form of the above is

$$\neg \forall x\, \exists y\ A_U(x, y) \leftrightarrow \forall f\, \exists x\ \neg A_U(x, fx).$$

Using this argument we would get $\neg(A^U) \leftrightarrow \forall f\, \exists x\ \neg A_U(x, fx)$, and the right-hand side of the equivalence is similar to the definition of $(\neg A)^U$. The difference is that the clause for negation has monotone quantifiers and includes the bounded quantifier $\widetilde{\exists} x' \trianglelefteq x$ so that any bound on an existential witness of A_U is itself an existential witness. In other words, the formula $A_U(x, y)$ is upward monotone in y.

Lemma 2.2 (Monotonicity)**.** *For any formula A of the language of $\mathsf{PA}^\omega_\trianglelefteq$, we have*

$$\mathsf{PA}^\omega_\trianglelefteq \vdash \forall x \; \forall y \; \forall y' \trianglelefteq y \; (A_U(x, y') \to A_U(x, y)).$$

Characteristic principles There are three principles that play an important role in the bounded functional interpretation, and in a sense characterise the mapping $A \mapsto A^U$ (see Theorem 2.14):

- the *monotone bounded choice* principle, denoted by mAC, is

$$\widetilde{\forall} x \; \widetilde{\exists} y \; A_{bd}(x, y) \to \widetilde{\exists} f \; \widetilde{\forall} x \; \widetilde{\exists} y \trianglelefteq fx \; A_{bd}(x, y),$$

  where $A_{bd}(x, y)$ is a bounded formula where f does not occur free, and x and y are tuples of variables of any type,

- the *bounded collection* principle, denoted by bC, is

$$\forall x \trianglelefteq a \; \exists y \; A_{bd}(x, y) \to \widetilde{\exists} b \; \forall x \trianglelefteq a \; \exists y \trianglelefteq b \; A_{bd}(x, y),$$

  where $A_{bd}(x, y)$ is a bounded formula where b does not occur free, and x and y are tuples of variables of any type,

- the *majorisability* principle, denoted by MAJ, is

$$\forall x \; \exists y \, (x \trianglelefteq y),$$

  where x is a variable of any type.

The monotone bounded choice principle is similar to the usual axiom of choice, but, instead of providing a choice function, it states the existence of a function that finds bounds for existential witnesses.

The bounded collection principle in the form presented above can also be used to get a bound for existential witnesses. Moreover, its contrapositive form

$$\widetilde{\forall} b \; \exists x \trianglelefteq a \; \forall y \trianglelefteq b \; A_{bd}(x, y) \to \exists x \trianglelefteq a \; \forall y \; A_{bd}(x, y)$$

can be used to prove the existence of an object $x \trianglelefteq a$ satisfying the global property $\forall y \, A_{bd}(x, y)$ from the weaker hypothesis that, for each bound b, there exist objects $x_b \trianglelefteq a$ satisfying the local property $\forall y \trianglelefteq b \, A_{bd}(x_b, y)$. The global object x is said to work *uniformly* for each bound b. This method can be used to prove weak König's lemma in $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{bC}$. The following example is from [12, §7].

Example 2.3. Weak König's lemma is the claim that every infinite binary tree has an infinite path. In $\mathsf{PA}^\omega_\trianglelefteq$, a tree is represented by a characteristic function T that, given a code s for a finite sequence, evaluates to $T(s) = 0$ if that finite sequence is in the tree, and to $T(s) = 1$ otherwise. Weak König's lemma corresponds to the formula

$$\forall T \trianglelefteq 1^{0\to 0} \; \big(\mathrm{Tree}_\infty(T) \to \exists x \trianglelefteq 1^{0\to 0} \; \forall k^0 \; T(x|k) = 0\big),$$

where $1^{0\to 0}$ is the constant function with value 1, the symbol $x|k$ denotes the code of the sequence $\langle x(0), \dots, x(k-1)\rangle$, and Tree_∞ is an abbreviation for the conjunction of

- $\forall r^0, s^0 \; \big(T(s) = 0 \wedge r \preceq s \to T(r) = 0\big)$ (tree)
- $\forall s^0 \; \big(T(s) = 0 \to \mathrm{Seq}_2(s)\big)$ (binary)
- $\forall n^0 \; \exists s^0 \; \big(T(s) = 0 \wedge |s| = n\big)$ (infinite)

where $r \preceq s$ means that the sequence encoded by r is an initial segment of the sequence encoded by s, the predicate $\mathrm{Seq}_2(s)$ says that s is the code of a binary sequence, and $|s|$ is the length of the sequence encoded by s.

The theory $\mathsf{PA}^{\omega}_{\trianglelefteq} + \mathsf{bC}$ proves weak König's lemma. Indeed, let $T \trianglelefteq 1^{0\to 0}$ and assume $\mathrm{Tree}_\infty(T)$. By the infinity property, for all n there is a code s for a sequence $\langle s(0), \ldots, s(n-1)\rangle$ of length n satisfying $T(s) = 0$. Define a functional x as

$$x(k) = \begin{cases} s(k) & \text{if } k < n \\ 0 & \text{otherwise.} \end{cases}$$

By the binary property, the sequence $\langle s(0), \ldots, s(n-1)\rangle$ is binary, hence $x \trianglelefteq 1^{0\to 0}$. Furthermore, by the tree property, we get $T(x|k) = 0$ for all $k \leq n$. We conclude

$$\forall n^0 \; \exists x \trianglelefteq 1^{0\to 0} \; \forall k \leq n \; T(x|k) = 0.$$

Using the contrapositive of bC, it follows that

$$\exists x \trianglelefteq 1^{0\to 0} \; \forall k^0 \; T(x|k) = 0.$$

This concludes the proof of weak König's lemma.

The contrapositive of the bounded collection principle bC can also be used to prove a principle of numerical comprehension for bounded formulas. The following argument is new, as far as we know, and was discovered in a meeting with Fernando Ferreira, whom we thank for the insight.

Example 2.4. The principle of *numerical comprehension for bounded formulas*, denoted by CA^0_{bd}, is

$$\exists f^{0\to 0} \; \forall n^0 \; (f(n) = 0 \leftrightarrow A_{bd}(n)),$$

where $A_{bd}(n)$ is a bounded formula where f does not occur free. As f is a characteristic function, we only need $f \trianglelefteq 1^{0\to 0}$, where $1^{0\to 0}$ is the constant function with value 1.

To prove this principle using bounded collection bC, we first prove

$$\forall a^0 \; \exists f \trianglelefteq 1^{0\to 0} \; \forall n \leq a \; (f(n) = 0 \leftrightarrow A_{bd}(n)),$$

by induction on a. The base case is $\exists f^{0\to 0} \, (f(0) = 0 \leftrightarrow A_{bd}(0))$. In the case that $A_{bd}(0)$ holds, take $f = \lambda k^0.0$. Otherwise, take $f = \lambda k^0.1$.

For the step case, we need a function $f \trianglelefteq 1^{0\to 0}$ such that

$$\forall n \leq a + 1 \; (f(n) = 0 \leftrightarrow A_{bd}(n)),$$

assuming that there is a function $g \trianglelefteq 1^{0\to 0}$ such that $\forall n \leq a \; (g(n) = 0 \leftrightarrow A_{bd}(n))$. In the case that $A_{bd}(a + 1)$ holds, define f taking the same values as g except $f(a + 1) = 0$. In the other case, define

$f(a+1) = 1$ instead.

Having proven $\forall a^0 \, \exists f \trianglelefteq 1^{0\to 0} \, \forall n \leq a \; (f(n) = 0 \leftrightarrow A_{bd}(n))$, we apply bounded collection to get

$$\exists f \trianglelefteq 1^{0\to 0} \; \forall n^0 \; (f(n) = 0 \leftrightarrow A_{bd}(n)).$$

Furthermore, applying the principle of bounded collection bC again, we can prove numerical comprehension for recursive formulas (the reason for the name recursive formula can be found in §3.4).

Example 2.5. The principle of *numerical comprehension for recursive formulas*, denoted by CA^0_Δ, is

$$\forall n^0 \, (\exists x^0 \, A_{bd}(n,x) \leftrightarrow \forall x^0 \, B_{bd}(n,x)) \to \exists f^{0\to 0} \, \forall n^0 \, (f(n) = 0 \leftrightarrow \exists x^0 \, A_{bd}(n,x)),$$

where $A_{bd}(n,x)$ and $B_{bd}(n,x)$ are bounded formulas where f does not occur free. As in the previous example, it suffices to have $f \trianglelefteq 1^{0\to 0}$, where $1^{0\to 0}$ is the constant function with value 1.

This argument is based on the proof of [16, Proposition 18]. Assume $\forall n^0 \, (\exists x^0 \, A_{bd}(n,x) \leftrightarrow \forall x^0 \, B_{bd}(n,x))$. We claim that

$$\forall a^0, b^0 \; \exists f \trianglelefteq 1^{0\to 0} \; \forall n \leq a \; \forall x \leq b \; \big((A_{bd}(n,x) \to f(n) = 0) \wedge (f(n) = 0 \to B_{bd}(n,x))\big).$$

(The bound $n \leq a$ is not necessary for the proof, but we write the claim in this form to apply bC) Indeed, by the principle CA^0_{bd} from Example 2.4, get a function f such that $f(n) = 0 \leftrightarrow \exists x \leq b \; A_{bd}(n,x)$. Now, let $n \leq a$ and $x \leq b$. We only need to check that if $f(n) = 0$, then $B_{bd}(n,x)$. In fact, if $f(n) = 0$, then $\exists x \, A_{bd}(n,x)$, and, by hypothesis, we get $\forall x \, B_{bd}(n,x)$. This proves the claim.

Using the contrapositive of bC on the previous claim, we get

$$\exists f \trianglelefteq 1^{0\to 0} \; \forall n^0, x^0 \; \big((A_{bd}(n,x) \to f(n) = 0) \wedge (f(n) = 0 \to B_{bd}(n,x))\big),$$

that is,

$$\exists f \trianglelefteq 1^{0\to 0} \; \forall n^0 \; \big((\exists x^0 \, A_{bd}(n,x) \to f(n) = 0) \wedge (f(n) = 0 \to \forall x^0 \, B_{bd}(n,x))\big),$$

and, by hypothesis, we conclude $\forall n^0 \, (f(n) = 0 \leftrightarrow \exists x^0 \, A_{bd}(n,x))$.

The majorisability principle MAJ is a generalisation of the following result.

Theorem 2.6 (Howard)**.** *For any closed term t of the language of $\mathsf{PA}^\omega_\trianglelefteq$, there is a closed term s such that $\mathsf{PA}^\omega_\trianglelefteq$ proves $t \trianglelefteq s$.*

The proof due to Howard in [23] in a different context still applies to the intensional theory $\mathsf{PA}^\omega_\trianglelefteq$. As this theorem only applies to closed terms, it cannot be used to prove MAJ in $\mathsf{PA}^\omega_\trianglelefteq$ — in fact, the theory $\mathsf{PA}^\omega_\trianglelefteq$ cannot prove MAJ, as the full set-theoretic model $\mathscr{S}^\omega$ contains functionals that cannot be strongly majorised.

There is an additional principle that can be used in place of monotone bounded choice — the *bounded choice* principle, denoted by bAC,

$$\forall x \; \exists y \; A_{bd}(x,y) \to \widetilde{\exists} f \; \widetilde{\forall} a \; \forall x \trianglelefteq a \; \exists y \trianglelefteq fa \; A_{bd}(x,y),$$

where $A_{bd}(x,y)$ is a bounded formula where f does not occur free, and x and y are tuples of variables of any type. The following result shows that monotone bounded choice and bounded choice are interchangeable.

Proposition 2.7. *The theory* $\mathsf{PA}^{\omega}_{\trianglelefteq} + \mathsf{bC}$ *proves* $\mathsf{mAC} \leftrightarrow \mathsf{bAC}$.

Proof. To prove $\mathsf{mAC} \to \mathsf{bAC}$, assume $\forall x \, \exists y \, A_{bd}(x,y)$. Then, it follows that $\widetilde{\forall} a \, \forall x \trianglelefteq a \, \exists y \, A_{bd}(x,y)$. By bounded collection, we get

$$\widetilde{\forall} a \; \widetilde{\exists} b \; \forall x \trianglelefteq a \; \exists y \trianglelefteq b \; A_{bd}(x,y).$$

Now use mAC to get

$$\widetilde{\forall} f \; \widetilde{\forall} a \; \widetilde{\exists} b \trianglelefteq fa \; \forall x \trianglelefteq a \; \exists y \trianglelefteq b \; A_{bd}(x,y).$$

By transitivity of $\trianglelefteq$, we get $\widetilde{\exists} f \; \widetilde{\forall} a \; \forall x \trianglelefteq a \; \exists y \trianglelefteq fa \, A_{bd}(x,y)$, the consequent of bAC.

To prove $\mathsf{bAC} \to \mathsf{mAC}$, assume $\widetilde{\forall} x \, \widetilde{\exists} y \, A_{bd}(x,y)$, i.e., $\forall x \, \exists y \, (x \trianglelefteq x \to y \trianglelefteq y \wedge A_{bd}(x,y))$. Using bAC, we get

$$\widetilde{\exists} f \; \widetilde{\forall} a \; \forall x \trianglelefteq a \; \exists y \trianglelefteq fa \; (x \trianglelefteq x \to y \trianglelefteq y \wedge A_{bd}(x,y)),$$

that is,

$$\widetilde{\exists} f \; \widetilde{\forall} a \; \widetilde{\forall} x \trianglelefteq a \; \widetilde{\exists} y \trianglelefteq fa \; A_{bd}(x,y).$$

As a is monotone, we can take $x = a$, yielding $\widetilde{\exists} f \, \widetilde{\forall} a \, \widetilde{\exists} y \trianglelefteq fa \, A_{bd}(a,y)$, the consequent of mAC. □

The theory $\mathsf{PA}^{\omega}_{\trianglelefteq}$ with the characteristic principles clashes with usual set-theoretic notions. This phenomenon is already apparent at the level of the axiom of extensionality. The problem stems from the uniformities introduced by the principle of bounded collection bC. In fact, consider the following example from [12].

Example 2.8. The theory $\mathsf{PA}^{\omega}_{\trianglelefteq} + \mathsf{bC}$ proves the negation of

$$\forall \Psi^{(0\to 0)\to 0} \; \forall f^{0\to 0}, g^{0 \to 0} \; (f = g \to \Psi f = \Psi g).$$

We argue by contradiction. For type $0 \to 0$, the formula $f = g$ is an abbreviation for $\forall k \, (fk = gk)$, so if the above was true we would have, in particular,

$$\forall \Psi \trianglelefteq 1^{(0\to 0)\to 0} \; \forall f, g \trianglelefteq 1^{0\to 0} \; \big(\forall k \, (fk = gk) \to \Psi f = \Psi g\big),$$

where $1^{0\to 0}$ and $1^{(0\to 0)\to 0}$ are terms representing the constant function with value 1. This is equivalent to having

$$\forall \Psi \trianglelefteq 1^{(0\to 0)\to 0} \; \forall f, g \trianglelefteq 1^{0\to 0} \; \exists k \; (fk = gk \to \Psi f = \Psi g),$$

and, using the principle of bounded collection bC, we would get

$$\exists n \; \forall \Psi \trianglelefteq 1^{(0\to 0)\to 0} \; \forall f, g \trianglelefteq 1^{0\to 0} \; \exists k \leq n \; (fk = gk \to \Psi f = \Psi g),$$

that is,

$$\exists n \; \forall \Psi \trianglelefteq 1^{(0\to 0)\to 0} \; \forall f, g \trianglelefteq 1^{0\to 0} \; \big(\forall k \leq n \, (fk = gk) \to \Psi f = \Psi g\big).$$

Take such an n and define Ψ such that

$$\Psi(h) = \begin{cases} 0 & \text{if } h(n+1) = 0 \\ 1 & \text{otherwise.} \end{cases}$$

Define also functions $f(k) = 0$ for all k, and

$$g(k) = \begin{cases} 1 & \text{if } k = n+1 \\ 0 & \text{otherwise.} \end{cases}$$

Then we have $\Psi \trianglelefteq 1^{(0\to0)\to0}$, $f, g \trianglelefteq 1^{0\to0}$ and $\forall k \leq n\,(fk = gk)$, but $\Psi f = 0$ and $\Psi g = 1$, a contradiction.

From this example, we conclude that the full set-theoretic model $\mathscr{S}^\omega$ and the model of strongly majorisable functionals $\mathscr{M}^\omega$ are not models of bC. Furthermore, the following example (also from [12]) shows that the flattening of bC is contradictory in the theory PA^ω.

Example 2.9. The theory $\mathsf{PA}^\omega + \mathsf{bC}^*$, where bC^* is the flattening of bC, is inconsistent. First, observe that the theory PA^ω proves

$$\forall f \leq^* 1^{0\to0}\ (\neg f \leq^* 0^{0\to0} \to \exists k\,(fk \neq 0)),$$

where $1^{0\to0}$ represents the constant function with value 1, and $0^{0\to0}$ represents the constant function with value 0. The proof of the same statement with intensional majorisability $\trianglelefteq$ cannot be carried out in $\mathsf{PA}^\omega_\trianglelefteq$, since we are using the contrapositive of the axiom

$$\forall u, v\,(u \leq^* v \to xu \leq^* yv \wedge yu \leq^* yv) \to x \leq^* y,$$

and $\mathsf{PA}^\omega_\trianglelefteq$ only has the corresponding rule.

Now, having $\forall f \leq^* 1^{0\to0}\,\exists k\,(\neg f \leq^* 0^{0\to0} \to fk \neq 0)$, we can use the flattening of bounded collection bC^* to conclude

$$\exists n\ \forall f \leq^* 1^{0\to0}\ \exists k \leq n\ (\neg f \leq^* 0^{0\to0} \to fk \neq 0),$$

that is,

$$\exists n\ \forall f \leq^* 1^{0\to0}\ (\exists k\,(fk \neq 0) \to \exists k \leq n\,(fk \neq 0)).$$

But this is refuted by the function $f \leq^* 1^{0\to0}$ such that

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

Soundness At first sight, it is not clear whether the theory $\mathsf{PA}^\omega_\trianglelefteq$ plus the characteristic principles is consistent. As a matter of fact, the previous examples cast doubt on the consistency of this theory. However, the following theorem proves the consistency of the theory extended with the characteristic principles relative to the consistency of finite-type Peano arithmetic.

Theorem 2.10 (Soundness)**.** *Let $A(z)$ be a formula of the language of $\mathsf{PA}^\omega_\trianglelefteq$ with free variables z. If*

$$\mathsf{PA}^\omega_\trianglelefteq + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} \vdash A(z),$$

then there are closed monotone terms t such that

$$\mathsf{PA}^\omega_\trianglelefteq \vdash \widetilde{\forall} w \; \forall z \trianglelefteq w \; \widetilde{\forall} x \; A(z)_U(x, twx).$$

Moreover, the terms t can be computed in an effective manner from such a proof of $A(z)$.

The consistency result comes from taking A to be the formula $0 = 1$. It would then follow that if $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ}$ is inconsistent, then $\mathsf{PA}^\omega_\trianglelefteq$ is inconsistent as well.

The terms t in the theorem above are called *bounding terms*. We could also write the conclusion of the theorem in the form

$$\mathsf{PA}^\omega_\trianglelefteq \vdash \widetilde{\forall} w \; \forall z \trianglelefteq w \; \widetilde{\forall} x \; \widetilde{\exists} y \trianglelefteq twx \; A(z)_U(x, y),$$

which is equivalent to the one above, by the monotonicity lemma (Lemma 2.2). The soundness theorem is proven by induction on the proof of $A(z)$, and detailed proofs can be found in [12, Theorem 2] for the classical setting and in [19, Theorem 1] for the intuitionistic setting.

The bounded functional interpretation can be easily extended with universal bounded sentences. In fact, consider adding a principle that is a sentence of the form $\forall a \, A_{bd}(a)$, with $A_{bd}(a)$ a bounded formula. Its interpretation $(\forall a \, A_{bd}(a))^U$ is $\widetilde{\forall} b \, \forall a \trianglelefteq b \, A_{bd}(a)$ and does not have variables quantified by $\widetilde{\exists}$. Thus the tuple of witnessing monotone terms provided by the soundness theorem is the empty tuple. Adding the principle $\forall a \, A_{bd}(a)$ to $\mathsf{PA}^\omega_\trianglelefteq$ we get our desideratum

$$\mathsf{PA}^\omega_\trianglelefteq + \forall a \; A_{bd}(a) \vdash \widetilde{\forall} b \; \forall a \trianglelefteq b \; A_{bd}(a),$$

as $\forall a \, A_{bd}(a)$ has no free variables and hence z is the empty tuple.

Theorem 2.11 (Soundness extended with univeral bounded sentences)**.** *Let $A(z)$ be a formula of the language of $\mathsf{PA}^\omega_\trianglelefteq$ with free variables z, and Δ be a set of universal bounded sentences. If*

$$\mathsf{PA}^\omega_\trianglelefteq + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \Delta \vdash A(z),$$

then there are closed monotone terms t, effectively obtainable from a proof of $A(z)$, such that

$$\mathsf{PA}^\omega_\trianglelefteq + \Delta \vdash \widetilde{\forall} w \; \forall z \trianglelefteq w \; \widetilde{\forall} x \; A(z)_U(x, twx).$$

The soundness theorem can be used to extract bounding terms for witnesses of $\forall\exists$-formulas. Such bounds are usually what is sought in a proof mining application.

Corollary 2.12 (Extraction)**.** *Let $A_{bd}(x, y)$ be a bounded formula with free variables x and y. If*

$$\mathsf{PA}^\omega_\trianglelefteq + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} \vdash \forall x \; \exists y \; A_{bd}(x, y),$$

then there are closed monotone terms t, effectively obtainable from a proof of $\forall x \, \exists y \, A_{bd}(x, y)$, such that

$$\mathsf{PA}^\omega_\trianglelefteq \vdash \widetilde{\forall} w \; \forall x \trianglelefteq w \; \exists y \trianglelefteq tw \; A_{bd}(x, y).$$

Proof. The result follows by applying the soundness theorem to the formula $\forall x\, \exists y\, A_{bd}(x, y)$. First, we have to compute the interpretation of $\forall x\, \neg\forall y\, \neg A_{bd}(x, y)$:

$$\begin{array}{rcl}(\neg A_{bd}(x,y))^U & \text{is} & \neg A_{bd}(x,y) \\ (\forall y\, \neg A_{bd}(x,y))^U & \text{is} & \widetilde{\forall} a\; \forall y \trianglelefteq a\; \neg A_{bd}(x,y) \\ (\neg\forall y\, \neg A_{bd}(x,y))^U & \text{is} & \widetilde{\exists} a\; \exists a' \trianglelefteq a\; \neg\forall y \trianglelefteq a'\; \neg A_{bd}(x,y) \\ (\forall x\, \neg\forall y\, \neg A_{bd}(x,y))^U & \text{is} & \widetilde{\forall} b\; \widetilde{\exists} a\; \forall x \trianglelefteq b\; \exists a' \trianglelefteq a\; \neg\forall y \trianglelefteq a'\; \neg A_{bd}(x,y),\end{array}$$

which can be simplified to $\widetilde{\forall} b\; \widetilde{\exists} a\; \forall x \trianglelefteq b\; \exists a' \trianglelefteq a\; \exists y \trianglelefteq a'\; A_{bd}(x, y)$. By the soundness theorem (Theorem 2.10), there is a term t such that

$$\mathsf{PA}^\omega_\trianglelefteq \vdash \widetilde{\forall} b\; \forall x \trianglelefteq b\; \exists a' \trianglelefteq tb\; \exists y \trianglelefteq a'\; A_{bd}(x, y),$$

and by transitivity of $\trianglelefteq$ we conclude $\mathsf{PA}^\omega_\trianglelefteq \vdash \widetilde{\forall} b \forall x \trianglelefteq b\, \exists y \trianglelefteq tb\; A_{bd}(x, y)$. □

A consequence of the soundness theorem is that the theory $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ}$ is conservative over $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{MAJ}$ with respect to $\forall\exists$-formulas.

Corollary 2.13 (Conservation). *Let $A_{bd}(x, y)$ be a bounded formula with free variables x and y. If*

$$\mathsf{PA}^\omega_\trianglelefteq + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} \vdash \forall x\; \exists y\; A_{bd}(x, y),$$

then $\mathsf{PA}^\omega_\trianglelefteq \vdash \widetilde{\forall} w\; \forall x \trianglelefteq w\; \exists y\; A_{bd}(x, y)$. Moreover, we get

$$\mathsf{PA}^\omega_\trianglelefteq + \mathsf{MAJ} \vdash \forall x\; \exists y\; A_{bd}(x, y).$$

Characterisation The characteristic principles characterise the bounded functional interpretation in the following sense.

Theorem 2.14 (Characterisation). *Let A be a formula of the language of $\mathsf{PA}^\omega_\trianglelefteq$. We have*

$$\mathsf{PA}^\omega_\trianglelefteq + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} \vdash A \leftrightarrow A^U.$$

The proof of this result can be done by induction on the formula A, for a proof in the classical case see [12, Theorem 3], and for the intuitionistic case see [19, Theorem 3].

The characterisation theorem confirms that the characteristic principles are only mAC, bC, and MAJ. Indeed, assume there is an extra principle P which we could join to $\mathsf{PA}^\omega_\trianglelefteq$ in addition to the existing characteristic principles and still have a result similar to the soundness theorem. Then there would exist a closed monotone term t such that $\mathsf{PA}^\omega_\trianglelefteq \vdash \widetilde{\forall} x\; P_U(x, tx)$. Hence we would have $\mathsf{PA}^\omega_\trianglelefteq \vdash P^U$. By the characterisation theorem, we would conclude that $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} \vdash P$, thus P would be superfluous, since it would be a consequence of the existent principles.

The characterisation theorem can also be used to provide a simpler interpretation for the conjunction $A \wedge B$. Using our definition, the interpretation of $A \wedge B$ would be computed as $(\neg(\neg A \vee \neg B))^U$ and becomes rather intricate and puzzling. Nevertheless, thanks to the characterisation theorem, within the theory $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ}$, the interpretation $(A \wedge B)^U$ is equivalent to $A \wedge B$, which in turn is equivalent to $A^U \wedge B^U$. If A^U is $\widetilde{\forall} x \, \widetilde{\exists} y \; A_U(x, y)$ and B^U is $\widetilde{\forall} u \, \widetilde{\exists} v \; B_U(u, v)$, then it follows that the interpretation of $A \wedge B$ is equivalent to

$$\widetilde{\forall} x, u \; \widetilde{\exists} y, v \; (A_U(x, y) \wedge B_U(u, v)).$$

If our language had the symbol $\wedge$ as primitive, then we could use the above to define $(A \wedge B)^U$ and the soundness theorem would still hold.

Chapter 3

Bar recursion

The principle of *bar recursion* is a principle of definition introduced by Spector [39] in an extension to the dialectica interpretation of Gödel. This extension can be used to prove the consistency of analysis, that is, of second-order arithmetic PA_2. To this principle of definition corresponds a principle of proof named *bar induction*. Bar recursion is to bar induction as ordinary recursion is to ordinary induction.

In this chapter, we present an extension of the bounded functional interpretation with bar recursive functionals and prove that this extension enables the interpretation of proofs using the principle of numerical comprehension.

- In Section 3.1, we describe the principle of definition by bar recursion, first in an informal setting and then in the context of finite-type arithmetic.
- In Section 3.2, we verify that the model of strongly majorisable functionals allows definitions by bar recursion, unlike the standard set-theoretic model.
- In Section 3.3, we introduce the axiom of bar induction, beginning with a definition in the meta-context. Afterwards, we present the principle of bar induction within finite-type arithmetic and prove that it holds in a relevant extension of this theory.
- In Section 3.4, we use bar induction to prove the principle of dependent choices and the principle of full numerical comprehension. These principles are then used to provide an interpretation of some subsystems of second-order arithmetic, the stronger subsystems requiring more bar recursors.

3.1 Bar recursion

In what follows, we will see that bar recursion is just a disguise for transfinite recursion on a well-founded tree. We begin by presenting bar recursion informally before adding it to the theory of finite-type Peano arithmetic. First, we need to establish some notation. By S^* we mean the set of finite sequences of elements of a set S, including the empty sequence $\langle\rangle$. Finite sequences are written as $s = \langle s(0), s(1), \ldots, s(k)\rangle$, with their elements referred to as $s(i)$. Given $s \in S^*$, we define $|s|$ as the length of s and, if $i \leq |s|$, then $s|i$ denotes the truncation of s to its first i elements. For an infinite sequence $x \in S^{\mathbb{N}}$, we also use the notation $x|i$ with the same meaning. The symbol $*$ is used for concatenation in

S^*. For our current purposes, we need a designated zero element $0 \in S$, and, when $s \in S^*$, we write $\overline{s}$ for the infinite sequence resulting from s by appending zeros, i.e.

$$\overline{s}(i) = \begin{cases} s(i) & \text{if } i < |s| \\ 0 & \text{otherwise.} \end{cases}$$

Definition 3.1 (Bar recursion)**.** To define a function $S^* \to T$ by bar recursion, we require

- $Y\colon S^{\mathbb{N}} \to \mathbb{N}$, a condition function,
- $F\colon S^* \to T$, a base function,
- $G\colon S^* \times T^S \to T$, a step function.

We say that a function $B\colon S^* \to T$ is defined by *bar recursion* on Y, F and G when

$$B(s) = \begin{cases} F(s) & \text{if } \exists i \leq |s| \left(Y(\overline{s|i}) \leq i\right) \\ G(s, \lambda w.B(s * \langle w \rangle)) & \text{otherwise.} \end{cases}$$

The version presented above is a variation by Ferreira [14] of the original definition by Spector, where the condition $\exists i \leq |s| \left(Y(\overline{s|i}) \leq i\right)$ replaces the original $Y(\overline{s}) \leq |s|$.

The scheme above might not assign a value $B(s)$ to all finite sequences $s \in S^*$. When $B(s)$ is defined by the step case G, its value can depend on the value of some $B(s * \langle w \rangle)$, which might in turn depend on the value of some $B(s * \langle w, z \rangle)$. If this chain of dependencies never reaches the base case F, then $B(s)$ will not be defined. The occurrence of this infinite recurrence depends solely on the condition function Y. We illustrate this with an example adapted from [14].

Example 3.2. Let $Y\colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ be defined as

$$Y(x) = \begin{cases} 0 & \text{if } \forall k \left(x(k) \neq 0\right) \\ i+1 & \text{if } x(i) = 0 \text{ and } \forall k < i \left(x(k) \neq 0\right), \end{cases}$$

and consider

$$B(s) = \begin{cases} 0 & \text{if } \exists i \leq |s| \left(Y(\overline{s|i}) \leq i\right) \\ B(s * \langle 1 \rangle) & \text{otherwise.} \end{cases}$$

Then $B(\langle\rangle)$ is not defined. In fact, if a finite sequence s does not contain zeros, then $Y(\overline{s}) = |s| + 1$. Thus, we would have $B(\langle\rangle) = B(\langle 1 \rangle) = B(\langle 1, 1 \rangle) = \ldots$, and never reach the base case.

Using Ferreira's version of bar recursion, there is a simple criterion for the condition function Y that guarantees bar recursion to define a total function. Consider the set

$$T_Y = \{s \in S^* \mid \forall i \leq |s| \left(Y(\overline{s|i}) > i\right)\}.$$

This set is a tree, since $s * t \in T_Y$ implies $s \in T_Y$. We say that T_Y is a *well-founded* tree if, for all infinite sequences $x \in S^{\mathbb{N}}$, there is i such that $x|i \notin T_Y$, or equivalently $Y(\overline{x|i}) \leq i$. The following result from [14] justifies bar recursive definitions by transfinite recursion on the tree T_Y.

Proposition 3.3. *Let $Y\colon S^{\mathbb{N}} \to \mathbb{N}$. The tree T_Y is well-founded if and only if there is a map h from T_Y to the ordinals such that, if s is a strict subsequence of t, then $h(t) < h(s)$.*

The height map h can be used to define B by transfinite recursion. First, set $B(s) = F(s)$, if $s \notin T_Y$. When $s \in T_Y$, we can set $B(s) = G(s, \lambda w.B(s * \langle w\rangle))$ as it only depends on values of B at lower height. Indeed, if $s * \langle w\rangle \in T_Y$, then $h(s * \langle w\rangle) < h(s)$. On the other hand, if $s * \langle w\rangle \notin T_Y$, then $B(s * \langle w\rangle)$ was already defined by F.

There are two conditions which imply that T_Y is well-founded:

- the *continuity* condition, usually associated with the intuitionistic setting

$$\forall x \in S^{\mathbb{N}}\ \exists i \in \mathbb{N}\ \forall y \in S^{\mathbb{N}}\ \big(x|i = y|i \to Y(x) = Y(y)\big),$$

  which is equivalent to having Y a continuous function from the product $S^{\mathbb{N}}$ to $\mathbb{N}$, where S and $\mathbb{N}$ are given the discrete topology,
- the *bounding* condition,

$$\forall x \in S^{\mathbb{N}}\ \exists n \in \mathbb{N}\ \forall i \in \mathbb{N}\ \big(Y(\overline{x|i}) \leq n\big).$$

The continuity condition implies the bounding condition, however the bounding condition does not imply continuity. In fact, in §3.2 we see that the model $\mathscr{M}^\omega$ of strongly majorisable functionals satisfies the bounding condition, but it contains functionals Y that are not continuous. The example from [3] is the function $Y\colon (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ such that

$$Y(x) = \begin{cases} 1 & \text{if } x(n) = 0 \text{ for all } n \\ 0 & \text{otherwise.} \end{cases}$$

On one hand, the function Y is strongly majorised by the constant function with value 1. On the other hand, the function Y is not continuous. To see why, let x be the constant sequence with value 0, and $i \in \mathbb{N}$. Then, there is a sequence y that matches x in the first i elements, but is different from x, for instance the sequence whose $(i+1)$-th element is 1 and whose other elements are all 0, and we get $Y(x) = 0$ and $Y(y) = 1$.

3.1.1 Bar recursion in finite-type arithmetic

Bar recursion is defined using finite sequences, which are not immediately available in finite-type Peano arithmetic. What is available are infinite sequences, in the form of objects of type $0 \to \sigma$, that can be truncated to generate finite sequences. A *finite sequence* with elements of type σ is represented by a pair of objects: one of type 0 and the other of type $0 \to \sigma$. The intended meaning is that the infinite sequence — the object of type $0 \to \sigma$ — is truncated at a natural number — the object of type 0 — corresponding to the length of the finite sequence.

An issue with the representation of finite sequences is that there are multiple pairs representing the same sequence — when a finite sequence of length n is represented as n, x; the values of $x(n), x(n+1), \ldots$ can be arbitrary. Therefore, we choose a *canonical representative* for this finite sequence — the pair $n, \overline{x|n}$ whose infinite sequence is extended with zeros. This definition requires an explanation of what we mean by zero at higher types: there are *zero terms* defined recursively for each type: 0^0 is the constant 0 and $0^{\sigma\to\tau}$ is $\lambda x^\sigma.0^\tau$. Given an infinite sequence x of type $0 \to \sigma$, the sequence $\overline{x|n}$ is thus defined as

$$(\overline{x|n})(i) = \begin{cases} x(i) & \text{if } i < n \\ 0^\sigma & \text{otherwise.} \end{cases}$$

The concatenation operation $*$ can also be defined in this context. In the general case, we define $\overline{x|n * y|m}$ as the infinite sequence with has the first n elements of x, followed by the first m elements of y, followed by zeros. Most regularly, we only append one element to a sequence and use the notation $\overline{x|n * \langle w\rangle}$ to mean the sequence with the first n elements of x, followed by w, and extended with zeros, that is

$$(\overline{x|n * \langle w\rangle})(i) = \begin{cases} x(i) & \text{if } i < n \\ w & \text{if } i = n \\ 0^\sigma & \text{otherwise.} \end{cases}$$

To add bar recursion to arithmetic in all finite types, we need additional arithmetical constants: the *bar recursors* $B_{\sigma,\tau}$, of type

$$\big((0 \to \sigma) \to 0\big) \to \big(0 \to (0 \to \sigma) \to \tau\big) \to \big(0 \to (0 \to \sigma) \to (\sigma \to \tau) \to \tau\big) \to \tau.$$

As for the recursors, we allow τ to be a tuple of types, thus allowing for *simultaneous* bar recursion. The types σ and τ will often be omitted and we denote the bar recursors simply by B.

Moreover, when including the bar recursors, we assume that the existing axioms apply also to the formulas of the extended language, together with the new axioms:

- bar recursor axioms, denoted by BR,
  $\exists i \le n\,\big(y(\overline{x|i}) \le i\big) \to \big(A[B(y,f,g,n,x)/z] \leftrightarrow A[f(n,\overline{x|n})/z]\big)$
  $\forall i \le n\,\big(y(\overline{x|i}) > i\big) \to \big(A[B(y,f,g,n,x)/z] \leftrightarrow A[g(n,\overline{x|n},\lambda w.B(y,f,g,Sn,\overline{x|n * \langle w\rangle}))/z]\big)$
  where $A(z)$ is an atomic formula.

These axioms mirror the usual definition by bar recursion (Definition 3.1). The bar recursor $B(y,f,g,n,x)$ is evaluated using the canonical representative of the sequence represented by n, x; and hence is independent of the particular choice of representative. The bar recursor axioms, like the combinator and recursor axioms, can be proven to hold for any formula $A(z)$.

The theory of Peano arithmetic in all finite types with bar recursion is denoted by $\mathsf{PA}^\omega + \mathsf{BR}$. If we are referring to the version with intensional majorisability, then we write $\mathsf{PA}^\omega_{\trianglelefteq} + \mathsf{BR}$.

3.2 The model of strongly majorisable functionals

Our main model for bar recursion — a model where functionals defined by bar recursion are always total functions — is the model $\mathscr{M}^\omega$ of strongly majorisable functionals, defined in 2.1. The fact that $\mathscr{M}^\omega$ is a model of PA^ω allows us to add some truths of $\mathscr{M}^\omega$ to PA^ω without incurring in contradiction. We will use this ahead, e.g., to prove within the theory that some bar recursors are monotone.

Our goal is to prove that $\mathscr{M}^\omega$ is a model of bar recursion. However, this claim is contingent on $\mathscr{M}^\omega$ having an interpretation for the bar recursors. A natural interpretation based on Definition 3.1 would be desirable, but this might not define a total function. Nevertheless, as $\mathscr{M}^\omega$ satisfies the bounding condition, the bar recursion scheme will always yield a total function, see §3.1.

Proposition 3.4. *$\mathscr{M}^\omega$ satisfies the bounding condition. That is, all $Y\colon M_{0\to\sigma} \to \mathbb{N}$ in $\mathscr{M}^\omega$ satisfy*

$$\forall x \in M_{0\to\sigma}\ \exists n \in \mathbb{N}\ \forall i \in \mathbb{N}\ \big(Y(\overline{x|i}) \le n\big).$$

Proof. Let Y and x be in $\mathscr{M}^\omega$. Then $Y \le^* Y^*$ and $x \le^* x^*$ for some Y^* and x^*. Take $n = Y^*(x^*)$. Because $\overline{x|i} \le^* x^*$ for any i, we get $Y(\overline{x|i}) \le Y^*(x^*) = n$. □

To check that $\mathscr{M}^\omega$ is indeed a model of bar recursion, we need a few lemmas about strong majorisability. For their proofs, the reader is referred to [28, §3.6].

Lemma 3.5. *If $x \le^* x^*$, then $x^* \le^* x^*$. Hence, if $x \le^* x^*$ and $x^* \in M_\tau^{M_\sigma}$, then $x^* \in M_{\sigma\to\tau}$.*

Lemma 3.6. *Let x and x^* be functions $M_{\sigma_1} \to \cdots \to M_{\sigma_k} \to M_\tau$. Then $x \le^* x^*$ if and only if we have*

$$xy_1 \dots y_k \le^* x^* y_1^* \dots y_k^* \quad \textit{and} \quad x^* y_1 \dots y_k \le^* x^* y_1^* \dots y_k^*,$$

for all $y_i, y_i^* \in M_{\sigma_i}$ such that $y_i \le^* y_i^*$, with $i = 1, \dots, k$.

Definition 3.7 (x^M)**.** Let $x \in M_\tau^{M_0}$, where $\tau = \tau_1 \to \cdots \to \tau_k \to 0$. We define

$$x^M \quad \text{as} \quad \lambda n^0, v_1^{\tau_1}, \dots, v_k^{\tau_k}. \max\{xiv_1 \dots v_k \mid i \le n\}.$$

Lemma 3.8. *Let $x, y \in M_\tau^{M_0}$. If we have $x(n) \le^* y(n)$ for all $n \in \mathbb{N}$, then*

$$x \le^* y^M \quad \textit{and} \quad x^M \le^* y^M.$$

In particular, if $x \le^* x^*$ and $y \le^* y^*$, then $x, y \le^* \max(x^*, y^*)$ and $\max(x, y) \le^* \max(x^*, y^*)$.

Lemma 3.9. *For each type τ, we have $M_\tau^{M_0} = M_{0\to\tau}$.*

Now we are ready to prove that $\mathscr{M}^\omega$ is a model of bar recursion. The proof uses a principle that Bezem [3] calls *classical bar induction*, that is, from

- $\forall x^{0\to\sigma}\ \exists n_0\ \forall n \ge n_0\ P(n, \overline{x|n})$
- $\forall n^0\ \forall x^{0\to\sigma}\ \big(\forall w^\sigma\ P(n+1, \overline{x|n * \langle w\rangle}) \to P(n, \overline{x|n})\big)$

conclude $\forall n^0\ \forall x^{0\to\sigma}\ P(n, \overline{x|n})$.

Lemma 3.10. *Classical bar induction holds in $\mathscr{M}^\omega$.*

Proof. We prove the contrapositive. Assume there are n_0 and x_0 such that $\neg P(n_0, \overline{x_0|n_0})$ and that

$$\forall n^0\, \forall x^{0\to\sigma}\ \big(\forall w^\sigma\ P(n+1, \overline{x|n * \langle w\rangle}) \to P(n, \overline{x|n})\big),$$

or equivalently,

$$\forall n^0\, \forall x^{0\to\sigma}\ \big(\neg P(n, \overline{x|n}) \to \exists w^\sigma\ \neg P(n+1, \overline{x|n * \langle w\rangle})\big).$$

Using dependent choice on the meta-level, there is a sequence x whose first n_0 elements coincide with those of x_0 and such that $\forall n \geq n_0 \, \neg P(n, \overline{x|n})$. Using Lemma 3.9, it follows that $x \in M_{0\to\sigma}$. This proves the negation of the first hypothesis of classical bar induction. □

The terms B^* defined below will be proven to be strong majorants for the bar recursors.

Definition 3.11 (B^p, B^*)**.** For each bar recursor $B_{\sigma,\tau}$, we define the following terms:

$$B^p_{\sigma,\tau} \quad \text{is} \quad \lambda y, f, g, n, x. B_{\sigma,\tau} y^m f^m g_f n x,$$
$$B^*_{\sigma,\tau} \quad \text{is} \quad \lambda y, f, g, n, x. (B_{\sigma,\tau} y^m f^m g_f)^M n x,$$

where $y^m(x) = y(x^M)$, $f^m(n, x) = f(n, x^M)$, and $g_f(n, x, v) = \max\big(f(n, x^M), g(n, x^M, v)\big)$.

By definition, the notation $(B_{\sigma,\tau} y f g)^M$ means $\lambda n^0, x^{0\to\sigma}. \max\{B_{\sigma,\tau} y f g i x \mid i \leq n\}$, i.e., computing the maximum of $B_{\sigma,\tau} y f g$ over all initial subsequences of x with length at most n.

The following theorem is the key to showing that the bar recursors can be strongly majorised.

Theorem 3.12. *Let $y \leq^* y^*$, $f \leq^* f^*$, and $g \leq^* g^*$ be of appropriate types for bar recursion, and r and s be infinite sequences. For all $n \in \mathbb{N}$, it holds in $\mathscr{M}^\omega$ that, if $r(k) \leq^* s(k)$ for all $k < n$, then*

$$B^p y^* f^* g^* n r \leq^* B^p y^* f^* g^* n s$$
$$B^p y f g n r \leq^* B^p y^* f^* g^* n s$$
$$B y f g n r \leq^* B^p y^* f^* g^* n s.$$

Proof. This proof is an adaptation of the proof of [28, Theorem 11.17] to our bar recursors. We use classical bar induction, with $P(n, s)$ being

$$\forall r^{0\to\sigma} \big(\forall k < n \, (rk \leq^* sk) \to B y f g n r, B^p y f g n r, B^p y^* f^* g^* n r \leq^* B^p y^* f^* g^* n s\big).$$

Hypothesis 1: $\forall s^{0\to\sigma} \, \exists n_0 \, \forall n \geq n_0 \; P(n, \overline{s|n})$. Let $s \in M_{0\to\sigma}$. Then there is $s^* \in M_\sigma^{M_0}$ such that $s \leq^* s^*$, and, by Lemma 3.9, we get $s^* \in M_{0\to\sigma}$. Let $n_0 = y^*(s^*)^M$, and take $n \geq n_0$. By Lemma 3.8, we have $(\overline{s|n})^M \leq^* (s^*)^M$, and, hence, as y^* is monotone,

$$(y^*)^m(\overline{s|n}) = y^*(\overline{s|n})^M \leq y^*(s^*)^M = n_0 \leq n.$$

Now let r be an infinite sequence such that $rk \leq^* sk$, for all $k < n$. As $\overline{r|n} \leq^* (\overline{s|n})^M$ and $(\overline{r|n})^M \leq^* (\overline{s|n})^M$ by Lemma 3.8, and as $y \leq^* y^*$, we get

$$n \geq y^*(\overline{s|n})^M \geq \begin{cases} y^*(\overline{r|n})^M = (y^*)^m(\overline{r|n}) \\ y(\overline{r|n})^M = y^m(\overline{r|n}) \\ y(\overline{r|n}). \end{cases}$$

Thus, using $f \leq^* f^*$ and Lemma 3.6,

$$B^p y^* f^* g^* ns = (f^*)^m(n, \overline{s|n}) = f^*(n, (\overline{s|n})^M) \geq^* \begin{cases} f^*(n, (\overline{r|n})^M) = B^p y^* f^* g^* nr \\ f(n, (\overline{r|n})^M) = B^p y f g n r \\ f(n, \overline{r|n}) = Byfgnr. \end{cases}$$

<u>Hypothesis 2</u>: $\forall n^0 \forall s^{0\to\sigma} \big(\forall w^\sigma\, P(n+1, \overline{s|n * \langle w\rangle}) \to P(n, \overline{s|n})\big)$. Let $n \in \mathbb{N}$, and r and s be infinite sequences such that $rk \leq^* sk$, for all $k < n$. Consider the following cases:

<u>Case 1</u>: exists $i \leq n$ such that $(y^*)^m(\overline{s|i}) \leq i$. Then, as $\overline{r|i} \leq^* (\overline{s|i})^M$ and $(\overline{r|i})^M \leq^* (\overline{s|i})^M$ by Lemma 3.8, we get

$$i \geq (y^*)^m(\overline{s|i}) = y^*(\overline{s|i})^M \geq \begin{cases} y^*(\overline{r|i})^M = (y^*)^m(\overline{r|i}) \\ y(\overline{r|i})^M = y^m(\overline{r|i}) \\ y(\overline{r|i}). \end{cases}$$

Hence, using $f \leq^* f^*$ and Lemma 3.6,

$$B^p y^* f^* g^* ns = (f^*)^m(n, \overline{s|n}) = f^*(n, (\overline{s|n})^M) \geq^* \begin{cases} f^*(n, (\overline{r|n})^M) = B^p y^* f^* g^* nr \\ f(n, (\overline{r|n})^M) = B^p y f g n r \\ f(n, \overline{r|n}) = Byfgnr. \end{cases}$$

<u>Case 2</u>: for all $i \leq n$ have $(y^*)^m(\overline{s|i}) > i$. Let $w \leq^* w^*$ be of type σ. Then, for all $k < n+1$,

$$(\overline{r|n * \langle w\rangle})(k) \leq^* (\overline{s|n * \langle w^*\rangle})(k) \quad \text{and} \quad (\overline{s|n * \langle w\rangle})(k) \leq^* (\overline{s|n * \langle w^*\rangle})(k).$$

Using $P(n+1, \overline{s|n * \langle w^*\rangle})$, we have

$$B^p y^* f^* g^*(n+1)(\overline{s|n * \langle w^*\rangle}) \geq^* \begin{cases} B^p y^* f^* g^*(n+1)(\overline{s|n * \langle w\rangle}) \\ B^p y^* f^* g^*(n+1)(\overline{r|n * \langle w\rangle}) \\ B^p y f g(n+1)(\overline{r|n * \langle w\rangle}) \\ Byfg(n+1)(\overline{r|n * \langle w\rangle}), \end{cases}$$

and, by Lemma 3.6, we conclude

$$\lambda w.B^p y^* f^* g^*(n+1)(\overline{s|n * \langle w\rangle}) \geq^* \begin{cases} \lambda w.B^p y^* f^* g^*(n+1)(\overline{r|n * \langle w\rangle}) \\ \lambda w.B^p y f g(n+1)(\overline{r|n * \langle w\rangle}) \\ \lambda w.Byfg(n+1)(\overline{r|n * \langle w\rangle}). \end{cases}$$

In this case, we have

$$\begin{aligned} B^p y^* f^* g^* ns &= g^*_{f^*}(n, \overline{s|n}, \lambda w.B^p y^* f^* g^*(n+1)(\overline{s|n * \langle w\rangle})) \\ &= \max\Big(f^*(n, (\overline{s|n})^M), g^*\big(n, (\overline{s|n})^M, \lambda w.B^p y^* f^* g^*(n+1)(\overline{s|n * \langle w\rangle})\big)\Big), \end{aligned}$$

thus, by Lemma 3.8, it follows that

$$B^p y^* f^* g^* ns \geq^* \begin{cases} \max\Big(f^*(n, (\overline{r|n})^M), g^*\big(n, (\overline{r|n})^M, \lambda w. B^p y^* f^* g^* (n+1)(\overline{r|n * \langle w \rangle})\big)\Big), f^*(n, (\overline{r|n})^M) \\ \max\Big(f(n, (\overline{r|n})^M), g\big(n, (\overline{r|n})^M, \lambda w. B^p y f g (n+1)(\overline{r|n * \langle w \rangle})\big)\Big), f(n, (\overline{r|n})^M) \\ g\big(n, \overline{r|n}, \lambda w. Byfg(n+1)(\overline{r|n * \langle w \rangle})\big), f(n, \overline{r|n}), \end{cases}$$

therefore

$$B^p y^* f^* g^* ns \geq^* \begin{cases} B^p y^* f^* g^* nr \\ B^p y f g n r \\ Byfgnr. \end{cases}$$

To conclude the proof, we use classical bar induction, that holds in $\mathscr{M}^\omega$ by Lemma 3.10, to get $\forall n^0 \, \forall x^{0 \to \sigma} \, P(n, \overline{x|n})$, which is equivalent to our goal $\forall n^0 \, \forall x^{0 \to \sigma} \, P(n, x)$. □

Corollary 3.13. *In $\mathscr{M}^\omega$ we have $B \leq^* B^*$.*

Proof. Let $y \leq^* y^*$, $f \leq^* f^*$, and $g \leq^* g^*$ be of appropriate types for bar recursion, and $r \leq^* s$ be infinite sequences. Using Theorem 3.12, we get that, for all $n \in \mathbb{N}$,

$$Byfgnr \leq^* B^p y^* f^* g^* ns, \quad B^p yfgnr \leq^* B^p y^* f^* g^* ns \quad \text{and} \quad B^p y^* f^* g^* nr \leq^* B^p y^* f^* g^* ns.$$

Applying Lemma 3.6, yields

$$Byfgn \leq^* B^p y^* f^* g^* n \quad \text{and} \quad B^p yfgn \leq^* B^p y^* f^* g^* n,$$

for all $n \in \mathbb{N}$, therefore, by Lemma 3.8, we get

$$Byfg \leq^* (B^p y^* f^* g^*)^M = B^* y^* f^* g^* \quad \text{and} \quad B^* yfg = (B^p yfg)^M \leq^* (B^p y^* f^* g^*)^M = B^* y^* f^* g^*.$$

Using again Lemma 3.6, we conclude $B \leq B^*$. □

Corollary 3.14. *$\mathscr{M}^\omega$ is a model of bar recursion, that is, a model of* $\mathsf{PA}^\omega + \mathsf{BR}$.

Proof. The bar recursors $B_{\sigma,\tau}$ are strongly majorised by $B^*_{\sigma,\tau}$. Hence $B_{\sigma,\tau}$ is in $\mathscr{M}^\omega$. □

3.3 Bar induction

Bar induction is a principle of proof stemming from the bar theorem of Brouwer's intuitionism [6]. There are multiple variants of bar induction, see e.g. [24]. The version of bar induction relevant to this work is called monotone bar induction. We first present bar induction informally — using the language of §3.1 — before presenting a formal version within finite-type Peano arithmetic — using the representation of finite sequences described in §3.1.1.

Definition 3.15 (Monotone bar induction)**.** Given formulas $P(s)$ and $Q(s)$, where s is intended to range over the elements of S^*, the principle of *monotone bar induction*, or simply BI, is

$$\text{Hyp1} \wedge \text{Hyp2} \wedge \text{Hyp3} \wedge \text{Hyp4} \to Q(\langle\rangle),$$

where

- Hyp1 is $\forall x \in S^{\mathbb{N}}\ \exists i \in \mathbb{N}\ P(x|i)$ (well-foundedness)
- Hyp2 is $\forall s \in S^*\ \forall i \le |s|\ \big(P(s|i) \to P(s)\big)$ (monotonicity)
- Hyp3 is $\forall s \in S^*\ \big(P(s) \to Q(s)\big)$ (base)
- Hyp4 is $\forall s \in S^*\ \big(\forall w \in S\ Q(s * \langle w \rangle) \to Q(s)\big)$ (step)

Monotone bar induction is classically valid, despite its origin in intuitionism. Indeed, assuming $\neg Q(\langle\rangle)$, we can use Hyp4 to get $w_0 \in S$ such that $\neg Q(\langle w_0 \rangle)$. Again, using Hyp4, we get $w_1 \in S$ such that $\neg Q(\langle w_0, w_1 \rangle)$. Iterating this process — using dependent choices on the meta-level — we get an infinite sequence $x \in S^{\mathbb{N}}$ such that $\forall i \in \mathbb{N}\ \neg Q(x|i)$. Then, we can use Hyp3 to conclude $\forall i \in \mathbb{N}\ \neg P(x|i)$, proving the negation of Hyp1. The monotonicity hypothesis Hyp2 is not needed in the classical case, but it is necessary in the intuitionistic version, see [14, §5].

The existence and uniqueness of functions defined by bar recursion can be proven using monotone bar induction, when the condition function Y is associated with a well-founded tree T_Y, that is, assuming

$$\forall x \in S^{\mathbb{N}}\ \exists i \in \mathbb{N}\ \big(Y(\overline{x|i}) \le i\big).$$

Let Y, F and G be of compatible types for a bar recursive definition. To prove existence, take $P(s)$ as $\exists i \le |s|\ \big(Y(\overline{s|i}) \le i\big)$ and $Q(s)$ as

$$\exists B\ \forall t \in S^*\ \Big(\big[P(s*t) \to B(s*t) = F(s*t)\big] \\ \wedge \big[\neg P(s*t) \to B(s*t) = G(s*t, \lambda w.B(s*t*\langle w \rangle))\big]\Big).$$

The well-foundedness hypothesis Hyp1 is equivalent to the tree T_Y being well-founded, and the monotonicity hypothesis Hyp2 also holds. For the base hypothesis Hyp3, assume $P(s)$. Then, for any $t \in S^*$, we have $P(s*t)$. Taking $B = F$, we prove $Q(s)$. Lastly, to prove the step hypothesis Hyp4, assume $\forall w \in S\ Q(s * \langle w \rangle)$, i.e., that, for each w, there is B_w such that, for all $t' \in S^*$,

$$B_w(s*\langle w \rangle * t') = \begin{cases} F(s*\langle w \rangle * t') & \text{if } P(s*\langle w \rangle * t') \\ G(s*\langle w \rangle * t', \lambda z.B_w(s*\langle w \rangle * t' * \langle z \rangle)) & \text{if } \neg P(s*\langle w \rangle * t'). \end{cases}$$

We want to prove $Q(s)$. When $t \ne \langle\rangle$, we can write $t = \langle w \rangle * t'$ for some t', hence we can take $B(s*t) = B_w(s*\langle w \rangle * t')$. For $t = \langle\rangle$, take

$$B(s) = \begin{cases} F(s) & \text{if } P(s) \\ G(s, \lambda w.B_w(s*\langle w \rangle)) & \text{if } \neg P(s). \end{cases}$$

By construction, we have $B_w(s*\langle w \rangle) = B(s*\langle w \rangle)$ for all w, thus we conclude $Q(s)$.

Using monotone bar induction, it follows that $Q(\langle\rangle)$, that is, that there exists a function B defined by bar recursion on Y, F and G.

To prove uniqueness, consider functions B and B', and a condition function Y with a well-founded tree. Take $P(s)$ as above and $Q(s)$ as $\forall t \in S^*\ B(s*t) = B'(s*t)$. Then Hyp3 and Hyp4 mean that, for all $s \in S^*$, we have equality $B(s*t) = B'(s*t)$ for all t, if either $P(s)$ holds, or there is equality

$B(s*t) = B'(s*t)$ for all $t \neq \langle\rangle$. When B and B' are defined by bar recursion on the same functions, the hypotheses Hyp3 and Hyp4 hold. Indeed, if $P(s)$ holds, then $P(s*t)$ also holds and

$$B(s*t) = F(s*t) = B'(s*t).$$

On the other hand, if $\neg P(s)$ holds and $B(s*t) = B'(s*t)$ for all $t \neq \langle\rangle$, then

$$B(s) = G(s, \lambda w.B(s*\langle w\rangle)) = G(s, \lambda w.B'(s*\langle w\rangle)) = B'(s),$$

since $B(s*\langle w\rangle) = B'(s*\langle w\rangle)$, for all $w \in S$.

By monotone bar induction, we conclude $Q(\langle\rangle)$, which means that $B(s) = B'(s)$ for all $s \in S^*$.

3.3.1 Bar induction in finite-type arithmetic

Now we turn our attention to formalising bar induction in the theory of Peano arithmetic in all finite types. We work with a simplified version of bar induction with the formula P being the same as Q.

Definition 3.16 (Simplified monotone bar induction)**.** Given a formula $P(n^0, x^{0\to\sigma})$, the principle of *simplified monotone bar induction*, named BI^-, is

$$\text{Hyp1} \wedge \text{Hyp2} \wedge \text{Hyp3} \to P(0, 0^{0\to\sigma}),$$

where

- Hyp1 is $\forall x^{0\to\sigma}\ \exists n^0\ P(n, \overline{x|n})$ (well-foundedness)
- Hyp2 is $\forall n^0\ \forall x^{0\to\sigma}\ \forall i \leq n\ \big(P(i, \overline{x|i}) \to P(n, \overline{x|n})\big)$ (monotonicity)
- Hyp3 is $\forall n^0\ \forall x^{0\to\sigma}\ \big(\forall w^\sigma\, P(n+1, \overline{x|n*\langle w\rangle}) \to P(n, \overline{x|n})\big)$ (step)

When we restrict $P(n, x)$ to be an existential bounded formula, this principle is called *existential simplified monotone bar induction* and denoted by $\mathsf{BI}^-_\exists$.

The principle $\mathsf{BI}^-_\exists$ is known to hold for the theory $\mathsf{PA}^\omega + \mathsf{BR}$, in the presence of quantifier-free choice, see [14].

Our goal is to prove that the theory $\mathsf{PA}^\omega_\trianglelefteq$ with bar recursion, the characteristic principles, and the universal bounded truths of $\mathscr{M}^\omega$ proves $\mathsf{BI}^-_\exists$. This is a known result, proven by Engrácia in [10] as a consequence of bar induction holding in the intuitionistic setting and using a negative translation. Here we prove the result directly for the classical case.

First, we prove a lemma from [32], showing that adding bar recursion to PA^ω entails that functions $Y\colon (0\to\sigma)\to 0$ will always produce a well-founded tree T_Y.

Lemma 3.17 (Kreisel's trick)**.** *The theory* $\mathsf{PA}^\omega + \mathsf{BR}$ *proves*

$$\forall Y^{(0\to\sigma)\to 0}\, \forall x^{0\to\sigma}\, \exists i^0\, \big(Y(\overline{x|i}) \leq i\big),$$

for any type σ.

Proof. Take Y and x of the types above. Define, by bar recursion, a function $B\colon 0 \to (0 \to \sigma) \to 0$ such that

$$B(n,s) = \begin{cases} 0 & \text{if } \exists i \leq n \left(Y(\overline{s|i}) \leq i\right) \\ 1 + B(n+1, \overline{s|n * \langle x(n) \rangle}) & \text{otherwise,} \end{cases}$$

and let $h(k) = B(k, \overline{x|k})$. Then,

$$h(k) = \begin{cases} 0 & \text{if } \exists i \leq k \left(Y(\overline{x|i}) \leq i\right) \\ 1 + h(k+1) & \text{otherwise.} \end{cases}$$

Now, we prove that $h(h(0)) = 0$ and, hence, by definition of h, it must be $\exists i \leq h(0) \left(Y(\overline{x|i}) \leq i\right)$.

Proceed by contradiction, assuming $h(h(0)) \neq 0$. In general, when $h(k) \neq 0$ and $i \leq k$, we have $h(0) = i + h(i)$. In particular, it follows that $h(0) = k + h(k)$. If we had $h(h(0)) \neq 0$, then we would get $h(0) = h(0) + h(h(0))$, that is, $h(h(0)) = 0$, a contradiction.

Therefore, we conclude $h(h(0)) = 0$ and $\exists i \leq h(0) \left(Y(\overline{x|i}) \leq i\right)$. ☐

As in [10], we use some semantic reasoning based on the model $\mathscr{M}^\omega$. More precisely, we use some sentences from the set $\Delta_{\mathscr{M}^\omega}$ of formulas whose flattening is true in $\mathscr{M}^\omega$. The process of flattening is essentially replacing the intensional symbol $\trianglelefteq$ with the extensional $\leq^*$.

Definition 3.18 ($\Delta_{\mathscr{M}^\omega}$)**.** We denote by $\Delta_{\mathscr{M}^\omega}$ the set of universal bounded sentences of the language of $\mathsf{PA}^\omega_\trianglelefteq$ whose flattening holds in $\mathscr{M}^\omega$.

Before proving our main result, we must clarify an important matter about the consistency of the theory $\mathsf{PA}^\omega_\trianglelefteq$ extended simultaneously with the characteristic principles and with the sentences of $\Delta_{\mathscr{M}^\omega}$. After all, Example 2.8 shows that the theory $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{bC}$ is incompatible with the axiom of extensionality, yet we are adding to this theory sentences which are true in the model $\mathscr{M}^\omega$, a model that satisfies the axiom of extensionality. To the rescue comes the extended soundness theorem.

Theorem 3.19 (Soundness for $\Delta_{\mathscr{M}^\omega}$)**.** *Let $A(z)$ be a formula of the language of $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{BR}$ with free variables z. If*

$$\mathsf{PA}^\omega_\trianglelefteq + \mathsf{BR} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \Delta_{\mathscr{M}^\omega} \vdash A(z),$$

then there are closed monotone terms t, effectively obtainable from a proof of $A(z)$, such that

$$\mathsf{PA}^\omega_\trianglelefteq + \mathsf{BR} + \Delta_{\mathscr{M}^\omega} \vdash \widetilde{\forall} w \, \forall z \trianglelefteq w \, \widetilde{\forall} x \; A(z)_U(x, twx).$$

Proof. Apply the soundness theorem extended with universal sentences (Theorem 2.11) taking $\Delta = \mathsf{BR} + \Delta_{\mathscr{M}^\omega}$. ☐

Using this theorem, if the theory $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{BR} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \Delta_{\mathscr{M}^\omega}$ was inconsistent, then $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{BR} + \Delta_{\mathscr{M}^\omega}$ would be inconsistent as well. It follows that the former theory is consistent, since $\mathscr{M}^\omega$ is a model for the latter, by Corollary 3.14.

Theorem 3.20 (Bar induction)**.** *The theory* $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{BR} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \Delta_{\mathscr{M}^\omega}$ *proves* $\mathsf{BI}^-_\exists$.

Proof. This proof is based on the original by Howard [22] and on the variations by Engrácia in [9, 10] and by Ferreira in [14].

Let $P(n^0, r^{0\to\sigma})$ be the formula $\exists a^\tau\, Q(n, r, a)$, with Q a bounded formula. Without loss of generality, we can take $Q(n, r, a)$ upward monotone in a in the sense that $a \trianglelefteq b$ and $Q(n, r, a)$ imply $Q(n, r, b)$. To see why, write $P(n, r)$ as $\exists b^\tau\, P_{bd}(n, r, b)$, with P_{bd} a bounded formula. By MAJ, this is equivalent to $\exists a^\tau\, \exists b \trianglelefteq a\; P_{bd}(n, r, b)$. Taking $Q(n, r, a)$ as $\exists b \trianglelefteq a\; P_{bd}(n, r, b)$ we get an upward monotone bounded formula.

Now assume the hypotheses of $\mathsf{BI}^-_\exists$:

- $\forall r^{0\to\sigma}\; \exists n^0\; \exists a^\tau\; Q(n, \overline{r|n}, a)$ (Hyp1)
- $\forall n^0\; \forall r^{0\to\sigma}\; \forall i \leq n\; \big(\exists a^\tau\, Q(i, \overline{r|i}, a) \to \exists b^\tau\, Q(n, \overline{r|n}, b)\big)$ (Hyp2)
- $\forall n^0\; \forall r^{0\to\sigma}\; \big(\forall w^\sigma\, \exists a^\tau\, Q(n+1, \overline{r|n * \langle w\rangle}, a) \to \exists b^\tau\, Q(n, \overline{r|n}, b)\big)$ (Hyp3)

By Proposition 2.7, the theory $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{mAC} + \mathsf{bC}$ proves bAC. In this proof, it will be useful to use bounded choice instead of monotone choice. In fact, we will now use bAC on the hypotheses of $\mathsf{BI}^-_\exists$ to get functionals that bound the existential witnesses.

Using bAC on Hyp1, we get monotone functionals $Y\colon (0 \to \sigma) \to 0$ and $H\colon (0 \to \sigma) \to \tau$ such that

$$\widetilde{\forall} s^{0\to\sigma}\; \forall r \trianglelefteq s\; \exists n \leq Ys\; \exists a \trianglelefteq Hs\;\; Q(n, \overline{r|n}, a),$$

using the monotonicity of Q, this is equivalent to

$$\widetilde{\forall} s^{0\to\sigma}\; \forall r \trianglelefteq s\; \exists n \leq Ys\;\; Q(n, \overline{r|n}, Hs). \tag{Hyp1'}$$

Using bAC on Hyp2, yields a monotone functional $F\colon 0 \to (0 \to \sigma) \to \tau \to \tau$ such that

$$\forall n^0\; \widetilde{\forall} s^{0\to\sigma}\; \forall r \trianglelefteq s\; \widetilde{\forall} c^\tau\; \forall a \trianglelefteq c\; \exists b \trianglelefteq F(n, s, c)\; \big(\exists i \leq n\; Q(i, \overline{r|i}, a) \to Q(n, \overline{r|n}, b)\big),$$

and, by monotonicity of Q, we get

$$\forall n^0\; \widetilde{\forall} s^{0\to\sigma}\; \forall r \trianglelefteq s\; \widetilde{\forall} c^\tau\; \forall a \trianglelefteq c\; \big(\exists i \leq n\; Q(i, \overline{r|i}, a) \to Q(n, \overline{r|n}, F(n, s, c))\big). \tag{Hyp2'}$$

Using bAC and the fact that Q is upward monotone, the formula $\forall w^\sigma\, \exists a^\tau\;\, Q(n+1, \overline{r|n * \langle w\rangle}, a)$ in the hypothesis Hyp3 is equivalent to $\widetilde{\exists} f^{\sigma\to\tau}\, \widetilde{\forall} z^\sigma\, \forall w \trianglelefteq z\; Q(n+1, \overline{r|n * \langle w\rangle}, fz)$, thus Hyp3 is equivalent to

$$\forall n^0\; \forall r^{0\to\sigma}\; \widetilde{\forall} f^{\sigma\to\tau}\; \big(\widetilde{\forall} z^\sigma\, \forall w \trianglelefteq z\; Q(n+1, \overline{r|n * \langle w\rangle}, fz) \to \exists b^\tau\, Q(n, \overline{r|n}, b)\big).$$

By applying bAC again, there is a monotone functional $G\colon 0 \to (0 \to \sigma) \to (\sigma \to \tau) \to \tau$ (we disregard the witness for z) such that

$$\forall n^0\; \widetilde{\forall} s^{0\to\sigma}\; \forall r \trianglelefteq s\; \widetilde{\forall} g^{\sigma\to\tau}\; \forall f \trianglelefteq g\; \exists b \trianglelefteq G(n, s, g)\; \big(\widetilde{\forall} z^\sigma\, \forall w \trianglelefteq z\; Q(n+1, \overline{r|n * \langle w\rangle}, fz) \to Q(n, \overline{r|n}, b)\big),$$

finally, as Q is monotone, we conclude

$$\forall n^0\; \widetilde{\forall} s^{0\to\sigma}\; \forall r \trianglelefteq s\; \widetilde{\forall} g^{\sigma\to\tau}\; \forall f \trianglelefteq g\; \big(\widetilde{\forall} z^\sigma\, \forall w \trianglelefteq z\; Q(n+1, \overline{r|n * \langle w\rangle}, fz) \to Q(n, \overline{r|n}, G(n, s, g))\big). \tag{Hyp3'}$$

Now, we use the B^p bar recursor of Definition 3.11 to define

$$B(n,s) = B^p\big(Y, \lambda n, s.F(n,s,H(\overline{s|i_0})^M), G, n, s\big),$$

where i_0 is the least $i \leq n$ such that $Y(\overline{s|i})^M \leq i$, if such i exists, and n otherwise. The value of i_0 can be effectively obtained as a function of n and s within the theory. Thus

$$B(n,s) = \begin{cases} F\big(n, (\overline{s|n})^M, H(\overline{s|i_0})^M\big) & \text{if } \exists i \leq n\, \big(Y(\overline{s|i})^M \leq i\big) \\ \max\Big(F\big(n, (\overline{s|n})^M, H(\overline{s|i_0})^M\big), G\big(n, (\overline{s|n})^M, \lambda w.B(n, \overline{s|n * \langle w\rangle})\big)\Big) & \text{otherwise,} \end{cases}$$

since $\overline{\overline{s|n}\,|i_0}$ is the sequence $\overline{s|i_0}$.

With this definition, we claim that, for all n of type 0 and infinite sequences r and s such that $r(k) \trianglelefteq s(k)$ for all $k < n$, we have

$$\widetilde{\forall} z^\sigma\ \forall w \trianglelefteq z\ Q(n+1, \overline{r|n * \langle w\rangle}, B(n+1, \overline{s|n * \langle z\rangle})) \to Q(n, \overline{r|n}, B(n, \overline{s|n})).$$

To prove this, we prove two sub claims corresponding to the cases of bar recursion.

<u>Claim 1a</u>: $\exists i \leq n\, \big(Y(\overline{s|i})^M \leq i\big) \to Q(n, \overline{r|n}, B(n, \overline{s|n}))$.

Let i_0 be the least $i \leq n$ such that $Y(\overline{s|i})^M \leq i$. By Lemma 3.8, we have $\overline{r|i_0} \trianglelefteq (\overline{s|i_0})^M$. By Hyp1', there is $n_0 \leq Y(\overline{s|i_0})^M$ such that $Q(n^0, \overline{\overline{r|i_0}\,|n_0}, H(\overline{s|i_0})^M)$. Because $n_0 \leq Y(\overline{s|i_0})^M \leq i_0 \leq n$, the sequence $\overline{\overline{r|i_0}\,|n_0}$ is actually $\overline{r|n_0}$. Thus

$$Q\big(n_0, \overline{r|n_0}, H(\overline{s|i_0})^M\big).$$

Using Hyp2', as $\overline{r|n} \trianglelefteq (\overline{s|n})^M$ by Lemma 3.8, as $H(\overline{s|i_0})^M$ is monotone, and as $n_0 \leq n$, we get

$$Q\big(n, \overline{r|n}, F(n, (\overline{s|n})^M, H(\overline{s|i_0})^M)\big),$$

which is $Q(n, \overline{r|n}, B(n, \overline{s|n}))$, since we are assuming $\exists i \leq n\, \big(Y(\overline{s|i})^M \leq i\big)$.

<u>Claim 1b</u>:

$$\forall i \leq n\, \big(Y(\overline{s|i})^M > i\big) \to \big(\widetilde{\forall} z^\sigma\ \forall w \trianglelefteq z\ Q(n+1, \overline{r|n * \langle w\rangle}, B(n+1, \overline{s|n * \langle z\rangle})) \to Q(n, \overline{r|n}, B(n, \overline{s|n}))\big).$$

Take $f = \lambda w.B(n+1, \overline{s|n * \langle w\rangle})$. We can prove that this functional is monotone using the following sentence from $\Delta_{\mathscr{M}^\omega}$

$$\widetilde{\forall} y, f, g\ \forall n^0\ \forall r^{0\to\sigma}, s^{0\to\sigma}\ \big(\forall k < n\, (rk \trianglelefteq sk) \to B^p yfgnr \trianglelefteq B^p yfgns\big),$$

whose flattening holds in $\mathscr{M}^\omega$ by Theorem 3.12. Also, we have $\overline{r|n} \trianglelefteq (\overline{s|n})^M$ by Lemma 3.8. Assuming $\widetilde{\forall} z^\sigma\ \forall w \trianglelefteq z\ Q(n+1, \overline{r|n * \langle w\rangle}, B(n+1, \overline{s|n * \langle z\rangle}))$, we can use Hyp3' to get $Q(n, \overline{r|n}, G(n, (\overline{s|n})^M, f))$, that is,

$$Q\big(n, \overline{r|n}, G(n, (\overline{s|n})^M, \lambda w.B(n+1, \overline{s|n * \langle w\rangle}))\big).$$

In this case we are assuming $\forall i \leq n\,\big(Y(\overline{s|i})^M > i\big)$, hence

$$G\big(n, (\overline{s|n})^M, \lambda w.B(n+1, \overline{s|n * \langle w\rangle})\big)$$
$$\trianglelefteq \max\Big(F\big(n, (\overline{s|n})^M, H(\overline{s|i_0})^M\big), G\big(n, (\overline{s|n})^M, \lambda w.B(n+1, \overline{s|n * \langle w\rangle})\big)\Big) = B(n, \overline{s|n}).$$

By monotonicity of Q, we conclude $Q(n, \overline{r|n}, B(n, \overline{s|n}))$. This concludes the proof of Claim 1b.

Claim 1: Joining these claims, it follows that, for all n of type 0, and for all infinite sequences r and s such that $r(k) \trianglelefteq s(k)$ for all $k < n$, we have

$$\widetilde{\forall} z^\sigma\ \forall w \trianglelefteq z\ Q(n+1, \overline{r|n * \langle w\rangle}, B(n+1, \overline{s|n * \langle z\rangle})) \to Q(n, \overline{r|n}, B(n, \overline{s|n})),$$

and using bAC there is a monotone functional $T\colon 0 \to (0 \to \sigma) \to (0 \to \sigma) \to \sigma$ such that, for all n of type 0 and infinite sequences u and v, and for all $r \trianglelefteq u$ and $s \trianglelefteq v$, if r and s are such that $r(k) \trianglelefteq s(k)$ for all $k < n$, then we have

$$\widetilde{\forall} z^\sigma \trianglelefteq T(n, u, v)\ \forall w \trianglelefteq z\ Q(n+1, \overline{r * \langle w\rangle}, B(n, \overline{s|n * \langle z\rangle})) \to Q(n, \overline{r|n}, B(n, \overline{s|n})).$$

Define a functional $x\colon 0 \to \sigma$ by ordinary recursion with

$$x(0) = T(0, 0^{0\to\sigma}, 0^{0\to\sigma}) \quad \text{and} \quad x(n+1) = T(n, (\overline{x|n})^M, (\overline{x|n})^M).$$

Claim 2: By induction on n, we can prove that, for all n of type 0, and for all infinite sequences r and s, if

$$\forall k < n\,(r(k) \trianglelefteq s(k) \trianglelefteq x(k)) \to Q(n, \overline{r|n}, B(n, \overline{s|n})),$$

then it follows that $Q(0, 0^{0\to\sigma}, B(0, 0^{0\to\sigma}))$.

Base. Taking $n = 0$, then we get $Q(0, \overline{r|0}, B(0, \overline{s|0})) \to Q(0, 0^{0\to\sigma}, B(0, 0^{0\to\sigma}))$, which is true since the sequences $\overline{r|0}$ and $\overline{s|0}$ coincide with $0^{0\to\sigma}$.

Step. Assume that, for all infinite sequences r and s,

$$\forall k < n+1\,(r(k) \trianglelefteq s(k) \trianglelefteq x(k)) \to Q(n+1, \overline{r|n+1}, B(n+1, \overline{s|n+1})).$$

By the induction hypothesis, it suffices to prove $\forall k < n\,(r(k) \trianglelefteq s(k) \trianglelefteq x(k)) \to Q(n, \overline{r|n}, B(n, \overline{s|n}))$ to conclude $Q(0, 0^{0\to\sigma}, B(0, 0^{0\to\sigma}))$.

Let r and s be infinite sequences such that $r(k) \trianglelefteq s(k) \trianglelefteq x(k)$ for all $k < n$. Then, for any $w \trianglelefteq z \trianglelefteq T(n, (\overline{s|n})^M, (\overline{s|n})^M)$, we get that, for all $k < n+1$,

$$(\overline{r|n * \langle w\rangle})(k) \trianglelefteq (\overline{s|n * \langle z\rangle})(k) \trianglelefteq x(k).$$

In fact, since $(\overline{s|n})^M \trianglelefteq (x|n)^M$ by Lemma 3.8, and as T is monotone, we have $z \trianglelefteq T(n, (\overline{s|n})^M, (\overline{s|n})^M) \trianglelefteq T(n, (\overline{x|n})^M, (\overline{x|n})^M) = x(n)$.

By the initial hypothesis of the step case, we have $Q(n+1, \overline{r|n * \langle w\rangle}, B(n+1, \overline{s|n * \langle z\rangle}))$. Moreover, by Claim 1, as $w \trianglelefteq z \trianglelefteq T(n, (\overline{s|n})^M, (\overline{s|n})^M)$ were arbitrary, and as $\overline{r|n} \trianglelefteq (\overline{s|n})^M$ and $\overline{s|n} \trianglelefteq (\overline{s|n})^M$, it follows that $Q(n, \overline{r|n}, B(n, \overline{s|n}))$.

This proves $\forall k < n\,(r(k) \trianglelefteq s(k) \trianglelefteq x(k)) \to Q(n, \overline{r|n}, B(n, \overline{s|n}))$, and, using the induction hypothesis, we conclude $Q(0, 0^{0\to\sigma}, B(0, 0^{0\to\sigma}))$, ending the proof of the step case.

To conclude the proof, use Kreisel's trick to get n_0 such that $Y(\overline{x|n_0})^M \leq n_0$, i.e., apply Lemma 3.17 to the functional $Y^m(x) = Y(x^M)$. Now we use Claim 2: let r and s be arbitrary infinite sequences such that $r(k) \trianglelefteq s(k) \trianglelefteq x(k)$ for all $k < n_0$. Then, by Lemma 3.8, we get $(\overline{s|n_0})^M \trianglelefteq (\overline{x|n_0})^M$, thus, as Y is monotone, we get $Y(\overline{s|n_0})^M \leq Y(\overline{x|n_0})^M \leq n_0$. Using Claim 1a, it follows that $Q(n_0, \overline{r|n_0}, B(n_0, \overline{s|n_0}))$. This proves

$$\forall k < n_0\, (r(k) \trianglelefteq s(k) \trianglelefteq x(k)) \to Q(n_0, \overline{r|n_0}, B(n_0, \overline{s|n_0})),$$

and using Claim 2, we conclude $Q(0, 0^{0\to\sigma}, B(0, 0^{0\to\sigma}))$. That is, we prove $\exists a^\tau\, Q(0, 0^{0\to\sigma}, a)$, i.e., $P(0, 0^{0\to\sigma})$, as we wanted. □

Throughout the previous argument, we have used sentences from $\Delta_{\mathscr{M}^\omega}$ at will and often implicitly. Now we draw explicitly the uses of $\mathscr{M}^\omega$.

- The essential use of $\Delta_{\mathscr{M}^\omega}$ is in the form of Theorem 3.12, through the sentence

$$\tilde{\forall} y, f, g\, \forall n^0\, \forall r^{0\to\sigma}, s^{0\to\sigma}\, \big(\forall k < n\, (rk \trianglelefteq sk) \to B^p yfgnr \trianglelefteq B^p yfgns\big).$$

- Lemma 3.8 is also used, in the form

$$\forall n^0\, \forall r^{0\to\sigma}, s^{0\to\sigma}\, \big(\forall k < n\, (rk \trianglelefteq sk) \to \overline{r|n} \trianglelefteq (\overline{s|n})^M \wedge (\overline{r|n})^M \trianglelefteq (\overline{s|n})^M\big).$$

- The theory $\mathsf{PA}^\omega_\trianglelefteq$ is not endowed with the axiom of extensionality, hence we use $\Delta_{\mathscr{M}^\omega}$ for substitutions of the form

$$\forall n^0\, \forall r^{0\to\sigma}, s^{0\to\sigma}\, \big(\forall k < n\, (rk = sk) \wedge A_{bd}[\overline{r|n}/z] \to A_{bd}[\overline{s|n}/z]\big),$$

  for bounded formulas $A_{bd}(z)$.

3.4 Numerical comprehension

The addition of bar recursion significantly strengthens the theory of finite-type arithmetic. In fact, Gödel [21] already proposes this addition to the theory T — a theory similar to the quantifier-free fragment of PA^ω — in the context of the dialectica interpretation. Taking up this suggestion, Spector [39] extends the classical version of the dialectica interpretation to interpret not only first-order arithmetic, but also analysis — that is, second-order arithmetic PA_2 — in the theory $\mathsf{T} + \mathsf{BR}$.

Likewise, the bounded functional interpretation extended with bar recursion can also interpret a theory much stronger than $\mathsf{PA}^\omega_\trianglelefteq$. In fact, this extension can be used to interpret proofs that require full numerical comprehension. In particular, it can also be used to interpret PA_2. The key to showing this lies in that the theory $\mathsf{PA}^\omega_\trianglelefteq$ with the characteristic principles and bar recursion proves the principle of numerical comprehension CA^0.

Definition 3.21 (CA^0)**.** The principle of *numerical comprehension*, denoted by CA^0, is

$$\exists f^{0\to 0}\, \forall n^0\, (f(n) = 0 \leftrightarrow A(n)),$$

where $A(n)$ can be any formula where f does not occur free.

Before proving that CA^0 holds in the theory $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{BR}$ with the characteristic principles, we first prove some related choice principles.

Definition 3.22 (DC^ω)**.** The principle of *dependent choices*, denoted by DC^ω, is

$$\forall x^\sigma\, \exists y^\sigma\, A(x,y) \to \forall w^\sigma\, \exists f^{0\to\sigma} \big(f(0) = w \wedge \forall n^0\, A(f(n), f(n+1))\big),$$

where $A(x,y)$ can be any formula where f does not occur free, and σ any tuple of types. When we restrict $A(x,y)$ to be a universal bounded formula, this principle is denoted by $\mathsf{DC}^\omega_\forall$.

Definition 3.23 ($\mathsf{AC}^{0,\omega}$)**.** The principle of *numerical choice*, denoted by $\mathsf{AC}^{0,\omega}$, is

$$\forall n^0\, \exists x^\sigma\, A(n,x) \to \exists f^{0\to\sigma}\, \forall n^0\, A(n, f(n)),$$

where $A(n,x)$ can be any formula where f does not occur free, and σ any tuple of types. When we restrict σ to be type 0, this principle is denoted by $\mathsf{AC}^{0,0}$.

Proposition 3.24.

1. $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{BI}^-_\exists \vdash \mathsf{DC}^\omega_\forall$,
2. $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \mathsf{DC}^\omega_\forall \vdash \mathsf{DC}^\omega$,
3. $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{DC}^\omega \vdash \mathsf{AC}^{0,\omega}$,
4. $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{AC}^{0,0} \vdash \mathsf{CA}^0$.

Proof. The proofs are based on similar proofs in [10, §4] and [14, §7].

1. Let $A(x,y)$ be a universal bounded formula and assume $\forall x^\sigma\, \exists y^\sigma\, A(x,y)$. Let w_0 be of type σ. Then there is w_1 of type σ such that $A(w_0, w_1)$.

We will use $\mathsf{BI}^-_\exists$ in the equivalent form

$$\neg P(0, 0^{0\to\sigma}) \wedge \text{Hyp2} \wedge \text{Hyp3} \to \neg\text{Hyp1},$$

taking $P(n^0, s^{0\to\sigma})$ as

$$\exists i \leq n\, \neg A\big(\overline{\langle w_0, w_1\rangle * s|n}\,(i), \overline{\langle w_0, w_1\rangle * s|n}\,(i+1)\big).$$

Since $A(x,y)$ is universal bounded, the formula $P(n,s)$ is existential bounded.

The formula $\neg P(0, 0^{0\to\sigma})$ is equivalent to $A(w_0, w_1)$. Moreover, as $P(n_0, s)$ implies $P(n,s)$ for all $n \geq n_0$, it follows that

$$\forall n^0\, \forall s^{0\to\sigma}\, \forall n_0 \leq n\, \big(P(n_0, \overline{s|n_0}) \to P(n, \overline{s|n})\big),$$

i.e., Hyp2 holds.

To prove the contrapositive of Hyp3, $\neg P(n, \overline{s|n}) \to \exists w^\sigma\, \neg P(n+1, \overline{s|n * \langle w\rangle})$, let n be of type 0 and s be an infinite sequence such that $\neg P(n, \overline{s|n})$, that is,

$$\forall i \leq n\, A\big(\overline{\langle w_0, w_1\rangle * s|n}\,(i), \overline{\langle w_0, w_1\rangle * s|n}\,(i+1)\big).$$

By the hypothesis $\forall x\, \exists y\, A(x,y)$, there is y such that $A\big(\overline{\langle w_0, w_1\rangle * s|n}\,(n+1), y\big)$. Hence, it follows that

$$\forall i \leq n+1\ A\big(\overline{\langle w_0, w_1\rangle * s|n * \langle y\rangle}\,(i), \overline{\langle w_0, w_1\rangle * s|n * \langle y\rangle}\,(i+1)\big),$$

i.e., $\neg P(n+1, \overline{s|n * \langle y\rangle})$, and we get $\exists w^\sigma\, \neg P(n+1, \overline{s|n * \langle w\rangle})$.

Now, using $\mathsf{BI}_\exists^-$ in the form above, we conclude ¬Hyp1, meaning that there exists an infinite sequence z of type $0 \to \sigma$ such that $\forall n^0\, \neg P(n, \overline{z|n})$. In other words,

$$\forall n^0\ \forall i \leq n\ A\big(\overline{\langle w_0, w_1\rangle * z|n}\,(i), \overline{\langle w_0, w_1\rangle * z|n}\,(i+1)\big).$$

Defining f of type $0 \to \sigma$ as

$$f(n) = \begin{cases} w_0 & \text{if } n = 0 \\ w_1 & \text{if } n = 1 \\ z(n-2) & \text{otherwise,} \end{cases}$$

we conclude that it is such that $f(0) = w_0$ and $\forall n^0\, A(f(n), f(n+1))$.

2. Now let $A(x,y)$ be an arbitrary formula and assume $\forall x^\sigma\, \exists y^\sigma\, A(x,y)$.

By the characterisation theorem for the bounded functional interpretation (Theorem 2.14), the formula $A(x,y)$ is equivalent to $\tilde{\forall} u^\rho\, \tilde{\exists} v^\tau\, A_U(x,y,u,v)$, where A_U is a bounded formula, and ρ and τ are appropriate types. Using mAC, this is equivalent to $\tilde{\exists} g^{\rho\to\tau}\, \tilde{\forall} u^\rho\, \tilde{\exists} v \trianglelefteq gu\ A_U(x,y,u,v)$. Thus our initial hypothesis $\forall x^\sigma\, \exists y^\sigma\, A(x,y)$ can be written equivalently as

$$\forall x^\sigma\ \exists y^\sigma\ \exists g^{\rho\to\tau}\ \big(g \trianglelefteq g \wedge \tilde{\forall} u^\rho\, \tilde{\exists} v \trianglelefteq gu\ A_U(x,y,u,v)\big),$$

that is $\forall x^\sigma\, \exists y^\sigma\, \exists g^{\rho\to\tau}\, B(x,y,g)$ for a universal bounded formula B.

Hence, we can use $\mathsf{DC}_\forall^\omega$ (in full rigour, we would have to add a universally quantified dummy variable of the same type as g) to obtain

$$\forall w^\sigma\ \forall h^{\rho\to\tau}\ \exists f^{0\to\sigma}\ \exists \psi^{0\to\rho\to\tau}\ \big(f(0) = w \wedge \psi(0) = h \wedge \forall n^0\, B(f(n), f(n+1), \psi(n+1)\big),$$

i.e.,

$$\begin{aligned}\forall w^\sigma\ \forall h^{\rho\to\tau}\ \exists f^{0\to\sigma}\ \exists \psi^{0\to\rho\to\tau}\ \big(&f(0) = w \wedge \psi(0) = h \\ &\wedge \forall n^0\, \big(\psi(n+1) \trianglelefteq \psi(n+1) \wedge \tilde{\forall} u^\rho\, \tilde{\exists} v \trianglelefteq \psi(n+1)u\ A_U(f(n), f(n+1), u, v)\big)\big).\end{aligned}$$

Thus, discarding the upper bound $v \trianglelefteq \psi(n+1)u$, we get

$$\forall w^\sigma\ \exists f^{0\to\sigma}\ \big(f(0) = w \wedge \forall n^0\, \tilde{\forall} u^\rho\, \tilde{\exists} v^\tau\ A_U(f(n), f(n+1), u, v)\big),$$

and using the characterisation theorem again, we conclude

$$\forall w^\sigma\ \exists f^{0\to\sigma}\ \big(f(0) = w \wedge \forall n^0\, A(f(n), f(n+1))\big).$$

3. Assume $\forall n^0\, \exists x^\sigma\, A(n,x)$. Then, we have

$$\forall n^0\ \exists k^0\ \exists x^\sigma\ (k = n+1 \wedge A(k,x)).$$

Take some x_0 such that $A(0, x_0)$. Using DC^ω (again we would have to add a universally quantified dummy variable with the same type as x), there are functions g of type $0 \to 0$ and f of type $0 \to \sigma$ such that $g(0) = 0$, $f(0) = x_0$, and

$$\forall n^0 \left(g(n+1) = g(n) + 1 \wedge A(g(n+1), f(n+1))\right).$$

Using induction, we can prove that, for all n, $g(n) = n$. Therefore, it follows that $\forall n^0 \, A(n+1, f(n+1))$. Together with $A(0, x_0)$, we conclude $\forall n^0 \, A(n, f(n))$.

4. Applying $\mathsf{AC}^{0,0}$ to the formula $\forall n^0 \, \exists k^0 \, (k = 0 \leftrightarrow A(n))$, we get $\exists f^{0 \to 0} \, \forall n^0 \, (f(n) = 0 \leftrightarrow A(n))$. $\square$

Corollary 3.25. *The theory* $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{BR} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \Delta_{\mathscr{M}^\omega}$ *proves* CA^0*.*

Using this result, the bounded functional interpretation can be used to interpret the theory of second-order arithmetic PA_2.

Second-order arithmetic Second-order arithmetic, denoted by PA_2, is the extension of first-order Peano arithmetic PA with *set variables* representing sets of natural numbers, usually denoted by capital letters, e.g. X, Y, Z. These variables are then related to numerical terms t by $t \in X$, representing the relation of belonging to a set.

The axioms of PA_2 include the axioms of PA, the full second-order induction scheme

$$A(0) \wedge \forall n \, (A(n) \to A(n+1)) \to \forall n \, A(n),$$

where $A(n)$ can be any formula, and the comprehension scheme

$$\exists X \, \forall n \, (n \in X \leftrightarrow A(n)),$$

where $A(n)$ is an arbitrary formula where X does not occur free. More details about PA_2 can be seen in [38, I§2], where it is denoted by Z_2.

The language of second-order arithmetic PA_2 can be embedded in the finite-type language of PA^ω by considering number variables as variables of type 0 and set variables as variables of type $0 \to 0$ representing the characteristic function of the set. Formulas $t \in X$ are translated to $X(t) = 0$ (we consider characteristic functions returning the value 0 for elements of the set). With this translation, the syntactic similarity between the comprehension scheme above and CA^0 is now explained — the principle of numerical comprehension CA^0 is the translation of the comprehension scheme of PA_2.

With the embedding of the language of PA_2 into the language of PA^ω, we can use the bounded functional interpretation for proofs in second-order arithmetic PA_2.

Theorem 3.26 (Soundness for PA_2)**.** *Let $A(z)$ be a formula of the language of* PA_2*. If*

$$\mathsf{PA}_2 \vdash A(z),$$

then there are closed monotone terms t, effectively obtainable from a proof of $A(z)$, such that

$$\mathsf{PA}^\omega_\trianglelefteq + \mathsf{BR} + \Delta_{\mathscr{M}^\omega} \vdash \widetilde{\forall} w \, \forall z \trianglelefteq w \, \widetilde{\forall} x \; A^\omega(z)_U(x, twx),$$

where $A^\omega(z)$ is the translation of $A(z)$ to the language of $\mathsf{PA}^\omega_\trianglelefteq$.

Proof. By Corollary 3.25, if PA_2 proves $A(z)$, then $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{BR} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \Delta_{\mathscr{M}^\omega}$ proves the translation $A^\omega(z)$. Then apply the soundness theorem for bar recursion (Theorem 3.19). □

Recursive comprehension The theory RCA (named after the recursive comprehension axiom) is a subsystem of second-order arithmetic PA_2 with the full second-order induction scheme, and where the comprehension scheme

$$\exists X\, \forall n\, (n \in X \leftrightarrow A(n))$$

is restricted to recursive formulas $A(n)$ where X does not occur free. A *recursive formula*, also known as a Δ^0_1 formula, is a formula of the form $\exists x\, A_{bd}(n,x)$ that is equivalent to a formula of the form $\forall x\, B_{bd}(n,x)$, with $A_{bd}(n,x)$ and $B_{bd}(n,x)$ bounded. Hence the recursive comprehension scheme is commonly written in the form

$$\forall n\, (\exists x\, A_{bd}(n,x) \leftrightarrow \forall x\, B_{bd}(n,x)) \to \exists X\, \forall n\, (n \in X \leftrightarrow \exists x\, A_{bd}(n,x)).$$

Using the terminology of the arithmetical hierarchy, a recursive formula is a Σ^0_1 formula that is equivalent to a Π^0_1 formula. The name recursive formula comes from the fact that the sets definable by recursive formulas are precisely the recursive sets — being defined by a Σ^0_1 formula means that the set is recursively enumerable, and being defined by a Π^0_1 formula means that its complement is also recursively enumerable. The subsystem RCA_0 of RCA, can formalise much of what is known as recursive mathematics, see I§8 and II of [38].

The correspondent of recursive comprehension in $\mathsf{PA}^\omega_\trianglelefteq$ is the following principle from Example 2.5.

Definition 3.27 (CA^0_Δ)**.** The principle of *numerical comprehension for recursive formulas*, denoted by CA^0_Δ, is

$$\forall n^0\, (\exists x^0\, A_{bd}(n,x) \leftrightarrow \forall x^0\, B_{bd}(n,x)) \to \exists f^{0\to 0}\, \forall n^0\, (f(n) = 0 \leftrightarrow \exists x^0\, A_{bd}(n,x)),$$

where $A_{bd}(n,x)$ and $B_{bd}(n,x)$ are bounded formulas where f does not occur free.

By Example 2.5, the theory $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{bC}$ proves CA^0_Δ. Therefore, at the second-order level, the bounded functional interpretation without any bar recursion provides an interpretation of WKL, the subsystem of PA_2 that extends RCA with weak König's lemma — an axiom stating that every infinite binary tree has an infinite path (see Example 2.3).

Theorem 3.28 (Soundness for WKL)**.** *Let $A(z)$ be a formula of the language of* PA_2. *If*

$$\mathsf{WKL} \vdash A(z),$$

then there are closed monotone terms t, effectively obtainable from a proof of $A(z)$, such that

$$\mathsf{PA}^\omega_\trianglelefteq \vdash \tilde{\forall} w\ \forall z \trianglelefteq w\ \tilde{\forall} x\ A^\omega(z)_U(x, twx),$$

where $A^\omega(z)$ is the translation of $A(z)$ to the language of $\mathsf{PA}^\omega_\trianglelefteq$.

Proof. By Example 2.5, if RCA proves $A(z)$, then $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ}$ proves the translation $A^\omega(z)$. Furthermore, by Example 2.3, the theory $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{bC}$ proves weak König's lemma. Thus, if weak König's lemma is used in the proof of $A(z)$, then we still have a proof of $A^\omega(z)$ in $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ}$. Now apply the soundness theorem (Theorem 2.10) to conclude the proof. $\square$

3.4.1 Arithmetical comprehension

The theory ACA (named after the arithmetical comprehension axiom) is a subsystem of second-order arithmetic PA_2 that is stronger than RCA. It includes the full second-order induction scheme, and the comprehension scheme

$$\exists X \, \forall n \, (n \in X \leftrightarrow A(n))$$

is restricted to arithmetical formulas $A(n)$ — formulas without set quantifiers — where X does not occur free. This restriction is equivalent to taking $A(n)$ as a Π^0_1 formula — a formula of the form $\forall x \, A_{bd}(n, x)$ with $A_{bd}(n, x)$ bounded — in which X does not occur free, see [38, Lemma III.1.3]. Much of ordinary mathematics can be formalised within the subsystem ACA_0 of ACA, more details can be seen in I§3 and III of [38].

In PA^ω, the arithmetical comprehension scheme corresponds thus to a similar restriction of the principle of numerical comprehension CA^0.

Definition 3.29 ($\mathsf{CA}^0_\forall$)**.** The principle of *arithmetical comprehension*, denoted by $\mathsf{CA}^0_\forall$, is

$$\exists f^{0\to 0} \, \forall n^0 \, (f(n) = 0 \leftrightarrow A(n)),$$

where $A(n)$ is a universal bounded formula where f does not occur free, and with unbounded quantifiers only of type 0.

Definition 3.30 ($\mathsf{AC}^{0,0}_\forall$)**.** The principle of *arithmetical choice* of type 0, denoted by $\mathsf{AC}^{0,0}_\forall$, is

$$\forall n^0 \, \exists x^0 \, A(n, x) \to \exists f^{0\to 0} \, \forall n^0 \, A(n, f(n)),$$

where $A(n)$ is a universal bounded formula where f does not occur free, and with unbounded quantifiers only of type 0.

To prove the principle of arithmetical comprehension $\mathsf{CA}^0_\forall$, we only need the bar recursor $B_{0,0}$.

Proposition 3.31. *The theory* $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{BR}^{0,0} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ}$ *proves* $\mathsf{CA}^0_\forall$*, where* $\mathsf{BR}^{0,0}$ *is the bar recursor axiom for* $B_{0,0}$.

Proof. This observation follows from the fact that $\mathsf{CA}^0_\forall$ can be proven using $\mathsf{AC}^{0,0}_\forall$. Indeed, let $A(n)$ be $\forall a^0 \, A_{bd}(n, a)$, with $A_{bd}(n, a)$ a bounded formula. It can be proven that $\forall n^0 \, \exists k^0 \, (k = 0 \leftrightarrow A(n))$, or equivalently that

$$\forall n^0 \, \exists k^0 \, \exists b^0 \, \big((k = 0 \wedge \forall a^0 \, A_{bd}(n, a)) \vee (k \neq 0 \wedge \neg A_{bd}(n, b))\big).$$

Applying $\mathsf{AC}^{0,0}_\forall$, we get functions f and g of type $0 \to 0$ such that

$$\forall n^0 \, \big((f(n) = 0 \wedge \forall a^0 \, A_{bd}(n, a)) \vee (f(n) \neq 0 \wedge \neg A_{bd}(n, g(n)))\big).$$

From this, it follows that $\forall n^0 \, (f(n) = 0 \leftrightarrow \forall a^0 \, A_{bd}(n, a))$.

Now we use the argument of Proposition 3.24 to check that only the bar recursor $B_{0,0}$ is needed to prove $\mathsf{AC}_\forall^{0,0}$. In the proof of 3.24.3, to prove $\mathsf{AC}_\forall^{0,0}$ we apply DC^ω to the formula

$$\forall n^0\, \exists k^0\, \exists x^0\, (k = n + 1 \wedge A(k, x)),$$

that is, we only use $\mathsf{DC}_\forall^\omega$ for type $\sigma = 0$. In turn, according to the proof of 3.24.1, to prove this instance of $\mathsf{DC}_\forall^\omega$, we use bar induction for sequences s of type $0 \to 0$ and a formula $P(n, s)$ of the form $\exists a^0\, Q(n, s, a)$ — that is, with an unbounded quantifier of type 0. Inspecting the proof of Theorem 3.20, we see that to prove bar induction for a formula $P(n^0, s^{0\to\sigma})$ of the form $\exists a^\tau\, Q(n, s, a)$ we only need the bar recursor $B_{\sigma,\tau}$. Therefore, we only require $B_{0,0}$ to prove $\mathsf{AC}_\forall^{0,0}$.

We conclude that the theory $\mathsf{PA}_\trianglelefteq^\omega$ with the characteristic principles and the bar recursor $B_{0,0}$ proves the principle of arithmetical comprehension $\mathsf{CA}_\forall^0$. □

Using the argument above, it follows that the bounded functional interpretation extended with bar recursion for type 0 can already provide an interpretation of the arithmetic comprehension scheme.

Theorem 3.32 (Soundness for ACA)**.** *Let $A(z)$ be a formula of the language of PA_2. If*

$$\mathsf{ACA} \vdash A(z),$$

then there are closed monotone terms t, effectively obtainable from a proof of $A(z)$, such that

$$\mathsf{PA}_\trianglelefteq^\omega + \mathsf{BR}^{0,0} + \Delta_{\mathscr{M}^\omega} \vdash \tilde{\forall} w\, \forall z \trianglelefteq w\, \tilde{\forall} x\; A^\omega(z)_U(x, twx),$$

where $A^\omega(z)$ is the translation of $A(z)$ to the language of $\mathsf{PA}_\trianglelefteq^\omega$, and $\mathsf{BR}^{0,0}$ is the bar recursor axiom for $B_{0,0}$.

Proof. By Proposition 3.31, if ACA proves $A(z)$, then $\mathsf{PA}_\trianglelefteq^\omega + \mathsf{BR}^{0,0} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \Delta_{\mathscr{M}^\omega}$ proves the translation $A^\omega(z)$. Then apply the soundness theorem for bar recursion (Theorem 3.19) with $\mathsf{BR}^{0,0}$ instead of BR. □

These soundness theorems for PA_2, for ACA, and for WKL shed light on the proof-theoretic strength added by the bar recursors to the theory $\mathsf{PA}_\trianglelefteq^\omega + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ}$ already at the second-order level. Hence, in the context of the bounded functional interpretation, they illustrate the strength of the axioms that can be used in a proof so that we are guaranteed to find terms that bound the existential witnesses, this strength varying with the bar recursors that can occur in said terms:

- if no bar recursion is added, then we are at the level of the theory WKL, a subsystem of PA_2 with recursive comprehension, and with weak König's lemma,
- if bar recursion is added only for type 0, then we can prove the principle of arithmetical comprehension $\mathsf{CA}_\forall^0$, hence we reach the level of ACA,
- if bar recursion is added for all finite types, then the theory proves the principle of full numerical comprehension CA^0, and is at the level of second-order arithmetic PA_2.

Chapter 4

Abstract type

Adding an *abstract type* means adding a new ground type that represents an abstract space, such as a ring, a metric space, or a normed space. Up to this point, we have only considered types built from the ground type 0, representing the natural numbers. By using an abstract type, we are no longer restricted to working with spaces that have to be encoded. The typical example are Polish spaces — complete separable metric spaces — whose standard representation is described in [28, §4]. In particular, we can consider spaces that are not countable or countably based.

The use of an abstract type with a functional interpretation was introduced by Kohlenbach in [27] for bounded metric spaces and later by Gerhardy and Kohlenbach in [20] for unbounded metric spaces, in the context of the monotone functional interpretation. For the bounded functional interpretation, abstract types were introduced by Engrácia in [9] for normed spaces, and by Engrácia and Ferreira in [11] for a general space. This construction extends the possibilities for proof mining — that is, for the extraction of bounds using a functional interpretation — as this method can be now applied to theorems about arbitrary spaces. Moreover, even for spaces that can be represented by natural numbers, this approach has the advantage that we can write clearer statements when working with the objects directly instead of through their representatives.

In this chapter, we develop an extension of the bounded functional interpretation simultaneously with bar recursive functionals and with an abstract type.

- In Section 4.1, we extend the theory of Peano arithmetic in all finite types with an abstract type, extending the notion of strong majorisability to the extended types as well.
- In Section 4.2, we illustrate the general theory of abstract types with concrete examples from mathematical practice.
- In Section 4.3, we present the bounded functional interpretation extended with an abstract type and its characteristic principles. We also state the soundness and characterisation theorems.
- In Section 4.4, we extend the bar recursor axioms to the new abstract type, and check that the extended model of strongly majorisable functionals is a model of these axioms.
- In Section 4.5, we generalise bar induction to the abstract type case and prove that this principle of proof holds in a suitable extension of finite-type arithmetic.

- In Section 4.6, we use bar induction to prove the principles of dependent choices and full numerical comprehension. In an application of these principles, we show that the bounded functional interpretation automatically completes metric spaces.

4.1 Extended arithmetic in all finite types

The theory of arithmetic in all finite types can be extended with a new type X, called an *abstract type*, that represents a mathematical structure other than the natural numbers. We consider the addition of a single ground type representing a mathematical structure X, through we could consider adding multiple ground types simultaneously. This structure X can be a ring, or a metric space, for example. Some concrete examples are presented in §4.2.

Types The *extended finite types* are defined recursively as follows:

- 0 and X are extended finite types — the *extended ground types*,
- if σ and τ are extended finite types, then $\sigma \to \tau$ is an extended finite type — an *extended higher type*.

The original finite types using only a single ground type 0 are called *arithmetical types*.

Language The *language* of extended finite-type arithmetic includes

- variables $x^\tau, y^\tau, z^\tau, \ldots$ of each extended type τ,
- logical constants $\Pi_{\sigma,\tau}$ and $\Sigma_{\rho,\sigma,\tau}$, for all extended types ρ, σ and τ,
- arithmetical constants 0, S, and $R_{\boldsymbol{\tau}}$, for all non-empty tuples of extended types $\boldsymbol{\tau}$,
- mathematical constants
  - 0_X of type X,
  - other new constants related to the structure X,
- binary predicate symbols $=_0$ between objects of type 0, and $=_X$ between objects of type X.

The constant 0_X witnesses the fact that X is non-empty, and it is used for the canonical representation of finite sequences. The other mathematical constants depend on the structure X at hand — for example, if X is a ring, then one of these constants could be $+$ of type $X \to X \to X$ that represents an additive operation; or, if X is a metric space, we could consider a constant d of type $X \to X \to R$ representing a metric, where R represents the type of the real numbers (more details in §4.2).

Besides equality between objects of type 0, we also have primitive equality for objects of type X. In some cases, such as when the abstract type represents the real numbers, we may also want to have a defined version of equality between objects of type X, in addition to primitive logical equality. As before, equality for higher types is defined pointwise:

$$t =_{\sigma\to\tau} s \quad \text{is defined as} \quad \forall x^\sigma \, (tx =_\tau sx),$$

for any extended types σ and τ.

Axioms The *extended theory of Peano arithmetic in all finite types*, denoted by $\mathsf{PA}^{\omega,X}$, is the extension of the theory PA^{ω} with *axioms* that are the universal closures of the following formulas:

- equality axioms for type X

  $x =_X x$

  $x =_X y \wedge A[x/w] \to A[y/w]$

  where $A(w)$ is an atomic formula,
- universal axioms related to the structure X.

The equality axioms for type X can be proven to hold for arbitrary formulas $A(w)$ in $\mathsf{PA}^{\omega,X}$, as before.

The additional axioms are contingent upon the structure X. When X is a ring, we might have a distributivity axiom $(x+y)+z = x+(y+z)$; or, when X is a metric space, we could add the symmetry axiom $d(x,y) = d(y,x)$. The axioms should be universal because universal axioms can be added to the theory without any changes to the proof of the soundness theorem (as in Theorem 2.11). Otherwise, if we consider axioms that are not universal, then we have to modify the proof of the soundness theorem accordingly, by presenting bounding terms for the bounded functional interpretations of the new axioms.

Standard model The *extended full set-theoretic model*, denoted by $\mathscr{S}^{\omega,X}$, is the intended model for extended finite-type arithmetic. The type structure $\mathscr{S}^{\omega,X}$ is defined recursively as

- $S_0 = \mathbb{N}$,
- $S_X = X$,
- $S_{\sigma\to\tau} = S_\tau^{S_\sigma}$, for all types σ and τ.

The function symbols are then interpreted canonically — the interpretation of the new constants depends on the structure X — so that $\mathscr{S}^{\omega,X}$ is a model of $\mathsf{PA}^{\omega,X}$.

Majorisability The *extended strong majorisability* relation, due to Gerhardy and Kohlenbach [20], is a relation between objects of an extended type and of its corresponding arithmetical type. To each extended type τ we assign an arithmetical type $\hat{\tau}$ that is obtained from τ by replacing every occurrence of type X with type 0 — that is, $\hat{0}$ and $\hat{X}$ are 0, and $\widehat{\sigma\to\tau}$ is $\hat{\sigma}\to\hat{\tau}$. We define, by simultaneous recursion on the type τ, the extended strong majorisability relation $\leq^*_\tau$ and the sets M_τ as follows:

- $n \leq^*_0 m$ is simply $n \leq m$ and we define $M_0 = \mathbb{N}$,
- $x \leq^*_X n$ depends on the structure X and must satisfy
  - if $x \leq^*_X n$ and $n \leq m$, then $x \leq^*_X m$,
  - for each $x \in X$ there is some $n \in \mathbb{N}$ such that $x \leq^*_X n$,

  and we define $M_X = X$,
- $x \leq^*_{\sigma\to\tau} y$ holds when $x \in M_\tau^{M_\sigma}$, $y \in M_{\hat{\tau}}^{M_{\hat{\sigma}}}$, and

$$\forall u \in M_\sigma\ \forall v \in M_{\hat{\sigma}}\ (u \leq^*_\sigma v \to xu \leq^*_\tau yv) \wedge \forall u,v \in M_{\hat{\sigma}}\ (u \leq^*_{\hat{\sigma}} v \to yu \leq^*_{\hat{\tau}} yv)$$

  and we define $M_{\sigma\to\tau} = \{x \in M_\tau^{M_\sigma} \mid x \leq^* x^*,\ \text{for some } x^* \in M_{\hat{\tau}}^{M_{\hat{\sigma}}}\}$, for all extended types σ and τ.

The intuitive meaning of $x \leq^* y$ is the same as before — that y is an upper bound for x, and y is monotone — but the strong majorants are exclusively arithmetical. The extended version coincides with the original version of strong majorisability in Bezem [3] when τ is an arithmetical type, hence we use the same symbol $\leq^*$ for both relations. The main properties of the original strong majorisability relation still hold for the extended strong majorisability relation (for a proof see [9, Lemma 27]):

- reflexivity for strong majorants: if $x \leq^* y$, then we have $y \leq^* y$,
- transitivity: if $x \leq^* y$ and $y \leq^* z$, then $x \leq^* z$.

The extended model of *strongly majorisable functionals*, denoted by $\mathscr{M}^{\omega,X}$, is now based on the new majorisability relation, the type structure for $\mathscr{M}^{\omega,X}$ being defined with the sets M_τ defined above.

The function symbols are interpreted as in the standard model $\mathscr{S}^{\omega,X}$, but we now need the new constants — the ones related to the structure X — to have interpretations that are strongly majorisable. In fact, to prove the soundness theorem (Theorem 4.5) in the abstract type case, for each added constant c of type τ we require also a closed *companion term* t_c of type $\hat{\tau}$ that is a strong majorant. Moreover, the companion term of 0^X must be 0.

Intensional majorisability The definition of extended strong majorisability can be adapted to an *extensional* definition of majorisability within $\mathsf{PA}^{\omega,X}$:

- $t \leq^*_0 s$ is defined as $t \leq_0 s$,
- $t \leq^*_X s$ is defined as $B_X(t,s)$ for some universal formula $B_X(x,n)$ such that
  - $\mathsf{PA}^{\omega,X} \vdash \forall x^X \, \forall n^0, m^0 \; (B_X(x,n) \land n \leq_0 m \to B_X(x,m))$,
  - $\mathsf{PA}^{\omega,X} \vdash \forall x^X \, \exists n^0 \; B_X(x,n)$,
- $t \leq^*_{\sigma\to\tau} s$ is defined as

$$\forall u^\sigma \; \forall v^{\hat{\sigma}} \; (u \leq^*_\sigma v \to tu \leq^*_\tau sv) \land \forall u^{\hat{\sigma}}, v^{\hat{\sigma}} \; (u \leq^*_{\hat{\sigma}} v \to su \leq^*_{\hat{\tau}} sv)$$

  for all extended types σ and τ, assuming that u and v do not occur in t or in s.

The *intensional* definition of majorisability, as before, replaces the non-universal axioms of extensional majorisability with corresponding rules. First, we add to the language new binary relation symbols $\trianglelefteq_\tau$ infixing between objects of type τ and of type $\hat{\tau}$. We also add universal bounded quantifiers $\forall x \trianglelefteq_\tau t$, and the existential version $\exists x \trianglelefteq_\tau t$ as an abbreviation for $\neg \forall x \trianglelefteq_\tau t \, \neg A$.

The *extended theory of Peano arithmetic in all finite types with intensional majorisability*, denoted by $\mathsf{PA}^{\omega,X}_\trianglelefteq$, is the extension of $\mathsf{PA}^{\omega,X}$ with its axioms applying also to the new formulas with the symbol $\trianglelefteq$ and the bounded quantifiers, and additionally with the universal closures of the formulas:

- bounded quantifier axioms
  $\forall x \trianglelefteq_\tau t \; A \leftrightarrow \forall x \, (x \trianglelefteq_\tau t \to A)$
  [$\exists x \trianglelefteq_\tau t \; A \leftrightarrow \exists x \, (x \trianglelefteq_\tau t \land A)$, a consequence of the abbreviation $\exists x \trianglelefteq_\tau t \; A$]
  where A is an arbitrary formula, the variable x does not occur in the term t, and τ is any type,

- extended strong majorisability axioms

  $n \trianglelefteq_0 m \leftrightarrow n \leq_0 m$

  $x \trianglelefteq_X n \to B_X(x,n)$

  $x \trianglelefteq_{\sigma\to\tau} y \to \forall u^\sigma \, \forall v^{\hat\sigma} \; (u \leq^*_\sigma v \to xu \leq^*_\tau yv) \wedge \forall u^{\hat\sigma}, v^{\hat\sigma} \; (u \leq^*_{\hat\sigma} v \to yu \leq^*_{\hat\tau} yv)$

  for all extended types σ and τ,
- universal bounded axioms related to the structure X,

and the extended strong majorisability rules

$$\frac{A_{bd} \to B_X(x,n)}{A_{bd} \to x \trianglelefteq_X n}$$

and

$$\frac{A_{bd} \wedge u \trianglelefteq_\sigma v \wedge u' \trianglelefteq_{\hat\sigma} v' \to tu \trianglelefteq_\tau sv \wedge su' \trianglelefteq_{\hat\tau} sv'}{A_{bd} \to t \trianglelefteq_{\sigma\to\tau} s}$$

where A_{bd} is a bounded formula, and the variables u, v, u', and v' do not occur free in A_{bd}, in t or in s.

For the advantage of having universal axioms, attained by the inclusion of the rules above, we pay the price of not having a deduction theorem for $\mathsf{PA}^{\omega,X}_{\trianglelefteq}$. Besides universal axioms for the structure X, we can also have universal bounded axioms for the same reason — using the argument for Theorem 2.11, we can add them to the soundness theorem for free.

The intensional theory $\mathsf{PA}^{\omega,X}_{\trianglelefteq}$ can prove the main properties of extended strong majorisability:

- $x \trianglelefteq y \to y \trianglelefteq y$,
- $x \trianglelefteq y \wedge y \trianglelefteq z \to x \trianglelefteq z$.

A simple proof by induction can be found in [9, Lemma 28].

Flattening The formulas of the intensional theory $\mathsf{PA}^{\omega,X}_{\trianglelefteq}$ can be mapped to formulas of the theory $\mathsf{PA}^{\omega,X}$ using the extensional version of strong majorisability $\leq^*$. This process is called *flattening* — the flattening of a formula A of $\mathsf{PA}^{\omega,X}_{\trianglelefteq}$ is the formula A^* of $\mathsf{PA}^{\omega,X}$ obtained from A by replacing the relation symbols $\trianglelefteq_\tau$ with $\leq^*_\tau$, and the occurrences of bounded quantifiers $\forall x \trianglelefteq_\tau t \; A$ with $\forall x \, (x \leq^*_\tau t \to A)$.

When the theory $\mathsf{PA}^{\omega,X}_{\trianglelefteq}$ proves a formula A, it follows that $\mathsf{PA}^{\omega,X}$ proves the flattening A^* simply by using the same proof as for the original formula A but instead of using the rules of the intensional $\trianglelefteq$ we use the defining axioms of the extensional relation $\leq^*$.

4.2 Examples of abstract types

In the previous section, we paint a very general picture with an abstract type representing some mathematical structure X, and with the language, axioms and majorisability relation of $\mathsf{PA}^{\omega,X}_{\trianglelefteq}$ varying with each case. Now we present concrete examples of structures X and how these are embedded in the theory of finite-type arithmetic.

Rings The language is extended with constants related to ring theory:

- 0^X and 1^X of type X, representing the additive and multiplicative identities,
- $+$ and $\cdot$ of type $X \to X \to X$, representing the additive and multiplicative operations,
- $-$ of type $X \to X$, representing the symmetric.

The constant 0^X is part of the language for any abstract type, but here it has the additional mathematical meaning of being the additive identity.

The standard axioms for rings do not require a symbol for the symmetric $(-)$, and include an axiom stating the existence of a symmetric element

$$\forall x\, \exists y\, (x + y = y + x = 0).$$

The problem is that this axiom is not universal, so we use the universal algebra approach and consider a constant for the symmetric $(-)$ as part of the language and universal axioms from [4, I§8.1] that are universal closure of the following formulas

- $x + (y + z) = (x + y) + z$,
- $x + y = y + x$,
- $x + 0^X = x$,
- $x + (-x) = 0^X$,
- $x \cdot (y \cdot z) = (x \cdot y) \cdot z$,
- $x \cdot 1^X = 1^X \cdot x = x$,
- $(x + y) \cdot z = (x \cdot z) + (y \cdot z)$,
- $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$,

where x, y, and z are of type X. If the abstract type represents a commutative ring, we add to this list

- $x \cdot y = y \cdot x$.

The majorisability relation for the ground type X is the trivial one — we have $x \leq^*_X n$ for any pair (x, n). In this case, we can replace the majorisability rule for $\trianglelefteq_X$ with the axiom $\forall x^X\, \forall n^0\, (x \trianglelefteq_X n)$. With this majorisability relation, quantifications over the elements of the ring $\forall x^X$ are bounded quantifications $\forall x \trianglelefteq_X 0$, effectively rendering the elements of the ring computationally empty. Now, the axiom of existence of symmetric would not be problematic after all, as its quantifiers could be replaced by bounded quantifiers.

The companion terms of the constants — the terms t_c such that $\mathsf{PA}^{\omega,X}_{\trianglelefteq}$ proves $c \trianglelefteq t_c$ — are zero terms

- the companion of 0^X and 1^X is 0,
- the companion of $+$ and $\cdot$ is $\lambda n^0, m^0.0$,
- the companion of $-$ is $\lambda n^0.0$.

The abstract type of a ring has been used in proof mining studies to analyse a suitable version of the theorem of commutative algebra saying that an element is nilpotent if and only if it is contained in every prime ideal, cf. [1, Proposition 1.8]. This analysis uses the bounded functional interpretation and is performed by Ferreira in [15].

Real numbers There are a myriad of ways to incorporate the real numbers into the theory of finite-type arithmetic. Some are based on representations of real numbers as functions $\mathbb{N} \to \mathbb{N}$: one of them uses rapidly converging Cauchy sequences of rational numbers, see [28, §4.1]; other uses the signed-digit representation, see [9, §4.1.1]. The real numbers can also be added as an abstract type, see [17]. Instead of the usual X, we use R to denote the abstract type of the real numbers.

The language is extended to encompass ordered domains with constants:

- 0^R and 1^R of type R, representing the real numbers 0 and 1,
- $+$ and $\cdot$ of type $R \to R \to R$, representing the sum and product of real numbers,
- $-$ of type $R \to R$, representing the symmetric,
- inv of type $0 \to R$, with $\operatorname{inv} n$, sometimes written as $\frac{1}{n+1}$, representing the inverse of $n + 1$,

and with a binary relation

- $\leq_R$ between terms of type R, representing the usual order relation.

Adding a binary relation can be seen as adding a constant that represents its characteristic function. In fact, we could equivalently add a constant $\leq_R$ such that $\leq_R (x, y) = 0$ when $x \leq y$ as real numbers, and $\leq_R (x, y) = 1$ otherwise. The companion term for the constant $\leq_R$ would then be $\lambda n^0, m^0.1$.

The axioms for real numbers extend those used for commutative rings with the universal closure of the following formulas:

- $x \cdot y = 0^R \to x = 0^R \lor y = 0^R$,
- $(n + 1) \cdot \frac{1}{n+1} = 1^R$,
- $x \leq y \to x + z \leq y + z$,
- $0^R \leq x \land 0^R \leq y \to 0^R \leq x \cdot y$,

where n is of type 0, and x, y, and z are of type R. In the second formula, it appears that we are multiplying a term of type 0 with a term of type R, an operation not defined in the theory. Instead, we define a function $(\cdot)^R$ of type $0 \to R$ by recursion with $(0)^R = 0^R$ and $(n + 1)^R = (n)^R + 1^R$ that intuitively returns the real number corresponding to a natural number. When the intended type is clear from context, we write n instead of $(n)^R$.

The majorisability relation for the ground type R is defined as

$$x \leq^*_R n \quad \text{when} \quad |x| \leq_R n,$$

where $|x|$ is a function of type $R \to R$ such that

$$|x| = \begin{cases} x & \text{if } 0^R \leq x \\ -x & \text{otherwise.} \end{cases}$$

The companion terms for the constants are thus:

- the companions of 0^R and 1^R are 0 and 1,
- the companions of $+$ and $\cdot$ are $\lambda n^0, m^0.n + m$ and $\lambda n^0, m^0, n \cdot m$,
- the companion of $-$ is $\lambda n^0.n$,
- the companion of inv is $\lambda n^0.1$.

For proof mining purposes, there is a problem with the symbols $=_R$ and $\leq_R$: they have the same complexity as the corresponding symbols for type 0. For instance, equality between real numbers, which should have complexity similar to equality between functions $\mathbb{N} \to \mathbb{N}$, is represented by a symbol $=_R$ with the same syntactic complexity as equality between natural numbers $=_0$. With this in mind, alongside the primitive logical symbols $=_R$ and $\leq_R$, we define symbols for equality $=_\mathbb{R}$ and for inequality $\leq_\mathbb{R}$ whose intended meaning is equality or inequality as real numbers:

$$\begin{aligned} x =_\mathbb{R} y \quad &\text{is defined as} \quad \forall n^0 \left(|x - y| \leq_R \frac{1}{n+1} \right), \\ x \leq_\mathbb{R} y \quad &\text{is defined as} \quad \forall n^0 \left(x \leq_R y + \frac{1}{n+1} \right), \\ x <_\mathbb{R} y \quad &\text{is defined as} \quad \exists n^0 \left(x + \frac{1}{n+1} \leq_R y \right). \end{aligned}$$

These defined symbols connect the abstract type R of the real numbers with the natural numbers, and they define the relations of equality and inequality as they are frequently used in analysis. Furthermore, the symbol $=_\mathbb{R}$ induces a sort of Arquimedian property for R.

The abstract type R with equality $=_\mathbb{R}$ and inequality $\leq_\mathbb{R}$ in the sense of the real numbers can be proven to satisfy the axioms of an ordered field. For instance, if x of type R is such that $0^R <_\mathbb{R} x$, then there is y of type R such that $x \cdot y =_\mathbb{R} 1^R$, see [17] for a proof.

The following examples involve the real numbers. In the literature, these examples are presented using representations for the real numbers such as the signed-digit representation or Cauchy sequences. In what follows, we illustrate how these abstract types could be defined using the abstract type of the real numbers defined above.

Bounded metric spaces A metric space X comprises a metric $d\colon X \times X \to \mathbb{R}$. To consider metric spaces with a real-valued metric, we require two abstract types: one is X, representing the metric space; and the other is R, representing the real numbers.

The language is extended with a single constant to accommodate metric spaces:

- d of type $X \to X \to R$, representing the metric.

The axioms for a metric space with diameter bounded by $b \in \mathbb{N}$ are the universal closures of the following formulas:

- $d(x, x) =_\mathbb{R} 0^R$,
- $d(x, y) =_\mathbb{R} d(y, x)$,
- $d(x, z) \leq_\mathbb{R} d(x, y) + d(y, z)$,
- $d(x, y) \leq_\mathbb{R} b$,

where x, y, and z are of type X. As $=_{\mathbb{R}}$ and $\leq_{\mathbb{R}}$ are defined by universal formulas, the axioms we consider are indeed universal.

When the metric space is bounded, the majorisability relation for the ground type X is the trivial one, with $x \leq^* n$ for any n. In this case, we can discard the rule for $\trianglelefteq_X$ and have an axiom $\forall x^X \, \forall n^0 \, (x \trianglelefteq_X n)$.

The companion term for the constant d that represents the metric is thus $\lambda n^0, m^0.b$.

As for the real numbers, the metric spaces include a defined equality $=_{\mathbb{X}}$ in the sense of metric space equality

$$x =_{\mathbb{X}} y \quad \text{is defined as} \quad d(x,y) =_{\mathbb{R}} 0^R.$$

The reason for this is twofold: on one hand equality between elements of the metric space should have higher complexity than equality between natural numbers, that is, it should not be primitive recursive. Unlike the primitive $=_X$, the mathematical equality $=_{\mathbb{X}}$ provides computational information. On the other hand, metric spaces have the axiom

$$\forall x \, \forall y \, (x = y \leftrightarrow d(x,y) = 0),$$

and we cannot postulate the closure of $d(x,y) =_{\mathbb{R}} 0^R \to x =_X y$, since it is not a universal formula, because $=_{\mathbb{R}}$ is defined by a universal formula.

Metric spaces The case of a general metric space X is treated in a similar way to the bounded case, but with a different majorisability relation.

The axioms are the same axioms used for a bounded metric space, except for the axiom

$$\forall x^X \, \forall y^X \, d(x,y) \leq_{\mathbb{R}} b$$

stating that the diameter is bounded.

The majorisability relation for the ground type X is defined now using the basepoint 0^X

$$x \leq^*_X n \quad \text{is defined as} \quad d(x, 0^X) \leq_{\mathbb{R}} n,$$

and the companion term for the constant d is now $\lambda n^0, m^0.n + m$.

Normed spaces A normed space is a vector space with a norm. We consider vector spaces over the real numbers, hence we have two abstract types: one is X for the elements of the normed space, and the other is R for the real numbers.

The language is extended with constants to deal with normed spaces:

- 0^X of type X, representing the zero vector,
- $+$ of type $X \to X \to X$, representing vector addition,
- $\cdot$ of type $R \to X \to X$, representing scalar multiplication,
- $-$ of type $X \to X$, representing the symmetric,
- $\|\cdot\|$ of type $X \to R$, representing the norm.

We consider as axioms the universal closures of the following formulas from [4, II§1.1] and [5, IX§3.3]:

vector spaces are axiomatised by

- $x + (y + z) = (x + y) + z$,
- $x + 0^X = x$,
- $x + (-x) = 0^X$,
- $x + y = y + x$,
- $a \cdot (x + y) = (a \cdot x) + (a \cdot y)$,
- $(a + b) \cdot x = (a \cdot x) + (b \cdot x)$,
- $a \cdot (b \cdot x) = (a \cdot b) \cdot x$,
- $1^R \cdot x = x$,

and the axioms for a norm are

- $\|0\| =_{\mathbb{R}} 0^R$,
- $\|a \cdot x\| =_{\mathbb{R}} |a| \cdot \|x\|$,
- $\|x + y\| \leq_{\mathbb{R}} \|x\| + \|y\|$

where a and b are of type R; and x, y, and z are of type X. The equalities between terms of type X are primitive equalities $=_X$.

The majorisability relation for the ground type X is defined in a way similar to the case of the real numbers:

$$x \leq^*_X n \quad \text{is defined as} \quad \|x\| \leq_{\mathbb{R}} n.$$

The companion terms for the constants are also similar:

- the companion of 0^X is 0,
- the companions of $+$ and $\cdot$ are $\lambda n^0, m^0.n + m$ and $\lambda n^0, m^0.n \cdot m$,
- the companion of $-$ and $\| \cdot \|$ is $\lambda n^0.n$.

A normed space can be seen as a metric space. Indeed, the function $d(x, y) = \|x - y\|$ defines a metric. Thus, a normed vector space is a metric space with 0^X as the basepoint. As for metric spaces, we consider a defined equality symbol

$$x =_{\mathbb{X}} y \quad \text{defined as} \quad \|x - y\| =_{\mathbb{R}} 0^R,$$

for which the usual norm axiom

$$\forall x \, (x =_{\mathbb{X}} 0^X \leftrightarrow \|x\| =_{\mathbb{R}} 0^R)$$

holds. This is not an axiom we could have for primitive equality, as the axiom $\|x\| =_{\mathbb{R}} 0^R \to x =_X 0^X$ is not universal, due to the definition of $=_{\mathbb{R}}$.

Theorems about metric spaces and normed vector spaces are frequently the topic of proof mining studies, and having these spaces presented abstractly allows for greater generality. For example, the Browder fixed-point theorem [7] is analysed using the bounded functional interpretation in [35, 18].

4.3 Extended bounded functional interpretation

The extension of the theory of finite-type Peano arithmetic with abstract types is only worthwhile if we can then set up a functional interpretation. In fact, we will see that the bounded functional interpretation of §2.3 accepts additional abstract types with virtually no changes. In this section, we present the bounded functional interpretation extended with an abstract type and directly for classical logic, as presented in [9, §4.3].

First, we define the usual abbreviations for the monotone quantifiers — quantifiers ranging over monotone objects x, that is objects such that $x \trianglelefteq x$ — as follows

- $\tilde{\forall} x\, A$ is an abbreviation for $\forall x\, (x \trianglelefteq x \to A)$,
- $\tilde{\exists} x\, A$ is an abbreviation for $\exists x\, (x \trianglelefteq x \wedge A)$,
- $\tilde{\forall} x \trianglelefteq t\; A$ is an abbreviation for $\forall x \trianglelefteq t\, (x \trianglelefteq x \to A)$,
- $\tilde{\exists} x \trianglelefteq t\; A$ is an abbreviation for $\exists x \trianglelefteq t\, (x \trianglelefteq x \wedge A)$.

As monotone objects are those strongly majorised by themselves, and strong majorants are of arithmetical type, it follows that a monotone object must be of arithmetical type — that is, a type built using only the symbols 0 and $\to$.

Definition 4.1 (Extended bounded functional interpretation)**.** To each formula A of the language of $\mathsf{PA}^{\omega,X}_{\trianglelefteq}$ we assign a formula A^U, called the *extended bounded functional interpretation* of A, which is of the form

$$\tilde{\forall} x\; \tilde{\exists} y\; A_U(x, y),$$

with $A_U(x, y)$ a bounded formula, and x and y are (possibly empty) tuples of variables. The formulas A^U and A_U are defined recursively as follows:

- if A is an atomic formula, then A^U and A_U are simply A.

For the remaining cases, let the interpretation of A be $\tilde{\forall} x\, \tilde{\exists} y\; A_U(x, y)$ and the interpretation of B be $\tilde{\forall} u\, \tilde{\exists} v\; B_U(u, v)$. Then:

- $(\neg A)^U$ is $\tilde{\forall} f\; \tilde{\exists} x\; \tilde{\exists} x' \trianglelefteq x\; \neg A_U(x', fx')$,
- $(A \vee B)^U$ is $\tilde{\forall} x, u\; \tilde{\exists} y, v\; (A_U(x, y) \vee B_U(u, v))$,
- $(\forall z\, A)^U$ is $\tilde{\forall} w, x\; \tilde{\exists} y\; \forall z \trianglelefteq w\; A_U(x, y)$,
- $(\forall z \trianglelefteq t\; A)^U$ is $\tilde{\forall} x\; \tilde{\exists} y\; \forall z \trianglelefteq t\; A_U(x, y)$.

The formula inside the unbounded $\tilde{\forall}\,\tilde{\exists}$ quantifiers of A^U is defined to be A_U.

The clauses of the definition above are the same as in Definition 2.1. As a matter of fact, the interpretation is focused on the variables bound by the monotone quantifiers $\tilde{\forall} x\, \tilde{\exists} y$ and these only have meaning when x and y are of arithmetical type. Hence the bounded functional interpretation requires no syntactic changes to its definition.

As before, bounded formulas A are invariant under the interpretation, with both A^U and A_U being the formula A. Furthermore, the extended interpretation still yields formulas $A_U(x, y)$ that are upward monotone in the variable y.

Lemma 4.2 (Monotonicity)**.** *For any formula A of the language of $\mathsf{PA}^{\omega,X}_{\trianglelefteq}$, we have*

$$\mathsf{PA}^{\omega,X}_{\trianglelefteq} \vdash \forall x\, \forall y\, \forall y' \trianglelefteq y\ (A_U(x,y') \to A_U(x,y)).$$

Characteristic principles The original characteristic principles are naturally extended to the language that includes an abstract type. The main difference is that we consider variables of extended type, except for those bound by monotone quantifiers — in what follows, variables bound by a monotone quantifier $\widetilde{\forall} x$ are of arithmetical type, and variables bound by the usual quantifiers $\forall x$ can be of any extended type. The characteristic principles of the extended interpretation are:

- the *monotone bounded choice* principle, denoted by mAC, is

  $$\widetilde{\forall} x\, \widetilde{\exists} y\ A_{bd}(x,y) \to \widetilde{\exists} f\, \widetilde{\forall} x\, \widetilde{\exists} y \trianglelefteq fx\ A_{bd}(x,y),$$

  where $A_{bd}(x,y)$ is a bounded formula where f does not occur free, and x and y are tuples of variables of arithmetical type,
- the *bounded collection* principle, denoted by bC, is

  $$\forall x \trianglelefteq a\ \exists y\ A_{bd}(x,y) \to \widetilde{\exists} b\, \forall x \trianglelefteq a\ \exists y \trianglelefteq b\ A_{bd}(x,y),$$

  where $A_{bd}(x,y)$ is a bounded formula where b does not occur free, and x and y are tuples of variables of any extended type,
- the *majorisability* principle, denoted by MAJ, is

  $$\forall x\, \exists y\, (x \trianglelefteq y),$$

  where x is a variable of any extended type.

We use the same names for the characteristic principles in the context of abstract types, but they now refer to more general schemata including extended types.

The majorisability principle MAJ now generalises the extension of Howard's theorem (Theorem 2.6).

Theorem 4.3 (Howard extended)**.** *For any closed term t of the language of $\mathsf{PA}^{\omega,X}_{\trianglelefteq}$, there is a closed term s such that $\mathsf{PA}^{\omega,X}_{\trianglelefteq}$ proves $t \trianglelefteq s$.*

This result can be proven by induction using the companion terms for the new constants, see [9, Lemma 31]. Later on, the existence of bounding terms is used to prove the extended soundness theorem (Theorem 4.5), hence the need for companion terms.

The principle of monotone bounded choice mAC can be substituted for the principle of bounded choice, denoted by bAC,

$$\forall x\, \exists y\ A_{bd}(x,y) \to \widetilde{\exists} f\, \widetilde{\forall} a\, \forall x \trianglelefteq a\ \exists y \trianglelefteq fa\ A_{bd}(x,y),$$

where $A_{bd}(x,y)$ is a bounded formula where f does not occur free, and x and y are tuples of variables of any extended type.

Proposition 4.4. *The theory $\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{bC}$ proves $\mathsf{mAC} \leftrightarrow \mathsf{bAC}$.*

Proof. The argument for Proposition 2.7 still applies in the presence of an abstract type. □

Soundness The soundness theorem proves the consistency of $\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ}$ relative to the consistency of $\mathsf{PA}^{\omega,X}_{\trianglelefteq}$, however, from the point of view of proof mining, the interest is in the existence of bounding terms t.

Theorem 4.5 (Soundness extended)**.** *Let $A(z)$ be a formula of the language of $\mathsf{PA}^{\omega,X}_{\trianglelefteq}$ with free variables z, and Δ be a set of universal bounded sentences. If*

$$\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \Delta \vdash A(z),$$

then there are closed monotone terms t, effectively obtainable from a proof of $A(z)$, such that

$$\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \Delta \vdash \widetilde{\forall} w\, \forall z \trianglelefteq w\, \widetilde{\forall} x\; A(z)_U(x, twx).$$

Proof. The full proof by induction on the proof of $A(z)$ can be found in [9, Theorems 28, 29, and 38]. We consider formal proofs use the proof calculus of Shoenfield [37, §8.3]. To clarify the requirement of companion terms for the new constants, we prove only the induction step corresponding to the substitution axiom $\forall u\, A(u) \to A(t)$, where t is a term free for x in A.

First, it is useful to compute the bounded functional interpretation of an implication $A \to B$, which is defined as $(\neg A \vee B)^U$. Assume that A^U is $\widetilde{\forall} x\, \widetilde{\exists} y\; A_U(x, y)$ and that B^U is $\widetilde{\forall} u\, \widetilde{\exists} v\; B_U(u, v)$. Then

$$\begin{aligned} (\neg A)^U \quad &\text{is} \quad \widetilde{\forall} f\; \widetilde{\exists} x\; \widetilde{\exists} x' \trianglelefteq x\; \neg A_U(x', fx') \\ (\neg A \vee B)^U \quad &\text{is} \quad \widetilde{\forall} f, u\; \widetilde{\exists} x, v\; \big(\widetilde{\exists} x' \trianglelefteq x\; \neg A_U(x', fx') \vee B_U(u, v)\big), \end{aligned}$$

that is equivalent to $\widetilde{\forall} f, u\; \widetilde{\exists} x, v\; \big(\widetilde{\forall} x' \trianglelefteq x\; A_U(x', fx') \to B_U(u, v)\big)$.

Now to prove the case of the substitution axiom $\forall u\, A(u) \to A(t)$, we need to compute its bounded functional interpretation:

$$\begin{aligned} (\forall u\, A(u))^U \quad &\text{is} \quad \widetilde{\forall} v, a\; \widetilde{\exists} b\; \forall u \trianglelefteq v\; A(u)_U(a, b), \\ A(t)^U \quad &\text{is} \quad \widetilde{\forall} c\; \widetilde{\exists} d\; A(t)_U(c, d), \\ (\forall u\, A(u) \to A(t))^U \quad &\text{is} \quad \widetilde{\forall} f, c\; \widetilde{\exists} v, a, d\; \big(\widetilde{\forall} v' \trianglelefteq v\; \widetilde{\forall} a' \trianglelefteq a\; \forall u \trianglelefteq v'\; A(u)_U(a', fv'a') \to A(t)_U(c, d)\big), \end{aligned}$$

Let z be a tuple with the free variables of A and the variables of t. We want to provide a tuple of closed monotone terms $t = (p, q, r)$ for the variables bound by the existential monotone quantifier $\widetilde{\exists} v, a, d$ such that $\mathsf{PA}^{\omega,X}_{\trianglelefteq}$ proves

$$\widetilde{\forall} w\; \forall z \trianglelefteq w\; \widetilde{\forall} f, c\; \big(\widetilde{\forall} v' \trianglelefteq pwfc\; \widetilde{\forall} a' \trianglelefteq qwfc\; \forall u \trianglelefteq v'\; A(u)_U(a', fv'a') \to A(t)_U(c, rwfc)\big).$$

The term $\lambda z.t[z]$ is a closed term, since all the variables of t are among z. By the extended Howard theorem (Theorem 4.3), there is a closed monotone term $\lambda z.s[z]$ such that $\lambda z.t[z] \trianglelefteq \lambda z.s[z]$. Hence, if $z \trianglelefteq w$, then $t[z] \trianglelefteq s[w]$. Take

$$\begin{aligned} p \quad &\text{as} \quad \lambda w, f, c.s[w], \\ q \quad &\text{as} \quad \lambda w, f, c.c, \\ r \quad &\text{as} \quad \lambda w, f, c.f(s[w])c, \end{aligned}$$

that are closed monotone terms. Then $\mathsf{PA}^{\omega,X}_{\trianglelefteq}$ proves

$$\widetilde{\forall} w\ \forall z \trianglelefteq w\ \widetilde{\forall} f, c\ \big(\widetilde{\forall} v' \trianglelefteq s[w]\ \widetilde{\forall} a' \trianglelefteq c\ \forall u \trianglelefteq v'\ A(u)_U(a', fv'a') \to A(t)_U(c, f(s[w])c)\big),$$

by taking $v' = s[w]$, $a' = c$, and $u = t$, as we wanted. □

The extended soundness theorem can also be applied to get bounding terms for witnesses of $\forall\exists$-formulas. This is the fundamental theorem for proof mining.

Corollary 4.6 (Extraction extended)**.** *Let $A_{bd}(x, y)$ be a bounded formula with free variables x and y. If*

$$\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} \vdash \forall x\ \exists y\ A_{bd}(x, y),$$

then there are closed monotone terms t, effectively obtainable from a proof of $\forall x\, \exists y\, A_{bd}(x, y)$, such that

$$\mathsf{PA}^{\omega,X}_{\trianglelefteq} \vdash \widetilde{\forall} w\ \forall x \trianglelefteq w\ \exists y \trianglelefteq tw\ A_{bd}(x, y).$$

Proof. Similar to the proof of Corollary 2.12. □

Moreover, we can use the result above to conclude that the theory $\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ}$ is conservative over $\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{MAJ}$ for $\forall\exists$-formulas.

Corollary 4.7 (Conservation)**.** *Let $A_{bd}(x, y)$ be a bounded formula with free variables x and y. If*

$$\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} \vdash \forall x\ \exists y\ A_{bd}(x, y),$$

then $\mathsf{PA}^{\omega,X}_{\trianglelefteq} \vdash \widetilde{\forall} w\ \forall x \trianglelefteq w\ \exists y\ A_{bd}(x, y)$. Moreover, we get

$$\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{MAJ} \vdash \forall x\ \exists y\ A_{bd}(x, y).$$

Characterisation In the presence of the characteristic principles, the theory of finite-type Peano arithmetic proves that every formula is equivalent to its bounded functional interpretation.

Theorem 4.8 (Characterisation)**.** *Let A be a formula of the language of $\mathsf{PA}^{\omega,X}_{\trianglelefteq}$. We have*

$$\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} \vdash A \leftrightarrow A^U.$$

This result can be proved by induction on the formula A, see [9, Theorem 39].

4.4 Bar recursion

The principle of definition by bar recursion can be extended to abstract types simply by allowing bar recursive definitions involving the extended types.

In the context of finite-type arithmetic, bar recursion deals with finite sequences represented by pairs: an object of type 0 representing the length of the sequence, and an object of type $0 \to \sigma$ representing an infinite sequence to truncate. Due to multiple pairs n, x representing the same sequence, we choose a canonical representative $n, \overline{x|n}$ with an infinite sequence that only has zeros after the truncation point.

Now, the definition of the *zero terms* is extended to encompass the new ground type: 0^0 is the constant 0, 0^X is the constant 0^X, and $0^{\sigma\to\tau}$ is the constant $\lambda x^\sigma.0^\tau$. As before, the truncated sequence $\overline{x|n}$ is defined as

$$(\overline{x|n})(i) = \begin{cases} x(i) & \text{if } i < n \\ 0^\sigma & \text{otherwise.} \end{cases}$$

The operation of concatenation, denoted by $*$, is used as previously defined in §3.1.1.

We add arithmetical constants to the language, the *bar recursors* $B_{\sigma,\tau}$, of type

$$\big((0\to\sigma)\to 0\big)\to\big(0\to(0\to\sigma)\to\tau\big)\to\big(0\to(0\to\sigma)\to(\sigma\to\tau)\to\tau\big)\to\tau,$$

where σ and τ can be any extended type. Furthermore, we have the usual axioms:

- bar recursor axioms, denoted by BR,
  $\exists i \le n\,\big(y(\overline{x|i}) \le i\big) \to \big(A[B(y,f,g,n,x)/z] \leftrightarrow A[f(n,\overline{x|n})/z]\big)$
  $\forall i \le n\,\big(y(\overline{x|i}) > i\big) \to \big(A[B(y,f,g,n,x)/z] \leftrightarrow A[g(n,\overline{x|n},\lambda w.B(y,f,g,Sn,\overline{x|n*\langle w\rangle}))/z]\big)$
  where $A(z)$ is an atomic formula.

These axioms evaluate the bar recursors using canonical representatives for the finite sequences, thus ensuring that their value does not change depending on the particular representative. The theory $\mathsf{PA}^{\omega,X}$ proves that the bar recursor axioms generalise for arbitrary formulas $A(z)$.

To denote the extended theory of finite-type Peano arithmetic with bar recursion we write $\mathsf{PA}^{\omega,X}+\mathsf{BR}$ for the extensional version, or $\mathsf{PA}^{\omega,X}_{\trianglelefteq}+\mathsf{BR}$ for the intensional version.

The extended model of strongly majorisable functionals The intended model for bar recursion with an abstract type is the model of strongly majorisable functionals $\mathscr{M}^{\omega,X}$, defined in §4.1. The possibility of a natural interpretation for the bar recursors — that is, one according to Definition 3.1 — is justified by the fact that $\mathscr{M}^{\omega,X}$ satisfies the bounding condition of §3.1, and hence definitions by bar recursion always yield a function defined for all inputs.

Proposition 4.9. *$\mathscr{M}^{\omega,X}$ satisfies the bounding condition. That is, all $Y\colon M_{\sigma\to 0}\to\mathbb{N}$ in $\mathscr{M}^{\omega,X}$ satisfy*

$$\forall x \in M_{\sigma\to 0}\ \exists n\in\mathbb{N}\ \forall i \in\mathbb{N}\ \big(Y(\overline{x|i})\le n\big).$$

Proof. The proof of Proposition 3.4 applies without any changes. □

The lemmas used to prove that the bar recursors are strongly majorisable in $\mathscr{M}^{\omega}$ can be generalised to the abstract type case. The proofs can be found in [28, §17.4].

Lemma 4.10. *If $x \le^* x^*$, then $x^*\le^* x^*$. Hence, if $x\le^* x^*$ and $x^*\in M_\tau^{M_\sigma}$, then $x^*\in M_{\sigma\to\tau}$.*

Lemma 4.11. *Let $x\colon M_{\sigma_1}\to\cdots\to M_{\sigma_k}\to M_\tau$ and $x^*\colon M_{\hat\sigma_1}\to\cdots\to M_{\hat\sigma_k}\to M_{\hat\tau}$. Then $x\le^* x^*$ if and only if we have*

$$xy_1\dots y_k \le^* x^*y_1^*\dots y_k^* \quad \textit{and} \quad x^*\hat y_1\dots\hat y_k\le^* x^*y_1^*\dots y_k^*,$$

for all $y_i\in M_{\sigma_i}$ and $\hat y_i, y_i^*\in M_{\hat\sigma_i}$ such that $y_i\le^* y_i^*$ and $\hat y_i\le^* y_i^*$, with $i=1,\dots,k$.

In what follows, the functionals $(\cdot)^M$ are only used for arithmetical types, but the definition can accept certain extended types.

Definition 4.12 (x^M)**.** Let $x \in M_\tau^{M_0}$, where $\tau = \tau_1 \to \cdots \to \tau_k \to 0$ and $\tau_1, \ldots, \tau_k$ are extended types. We define

$$x^M \quad \text{as} \quad \lambda n^0, v_1^{\tau_1}, \ldots, v_k^{\tau_k}. \max\{xiv_1 \ldots v_k \mid i \leq n\}.$$

Lemma 4.13. *Let $x \in M_\tau^{M_0}$ and $y \in M_{\hat{\tau}}^{M_0}$, where $\tau = \tau_1 \to \cdots \to \tau_k \to 0$. If we have $x(n) \leq^* y(n)$ for all $n \in \mathbb{N}$, then*

$$x \leq^* y^M \quad \text{and} \quad x^M \leq^* y^M.$$

In particular, if $x \leq^* x^*$ and $y \leq^* y^*$, then $x, y \leq^* \max(x^*, y^*)$ and $\max(x, y) \leq^* \max(x^*, y^*)$.

If τ is an arbitrary extended type, then we can only conclude that, if $x(n) \leq^* y(n)$ for all $n \in \mathbb{N}$, then $x \leq^* y^M$; and that when $x \leq^* x^*$ and $y \leq^* y^*$, we also have $x, y \leq^* \max(x^*, y^*)$.

Lemma 4.14. *For each extended type τ, we have $M_\tau^{M_0} = M_{0 \to \tau}$.*

As we had for $\mathscr{M}^\omega$, the principle of classical bar induction is valid in $\mathscr{M}^{\omega,X}$ and can be used to show that $\mathscr{M}^{\omega,X}$ is a model of bar recursion. This principle says that from the hypotheses

- $\forall x^{0\to\sigma}\ \exists n_0\ \forall n \geq n_0\ P(n, \overline{x|n})$
- $\forall n^0\ \forall x^{0\to\sigma}\ \big(\forall w^\sigma\ P(n+1, \overline{x|n * \langle w\rangle}) \to P(n, \overline{x|n})\big)$

we can conclude $\forall n^0\ \forall x^{0\to\sigma}\ P(n, \overline{x|n})$.

Lemma 4.15. *Classical bar induction holds in $\mathscr{M}^{\omega,X}$.*

Proof. The reasoning for Lemma 3.10 applies to $\mathscr{M}^{\omega,X}$ as well. □

The terms B^p and B^* are defined as before — since these are used as strong majorants, we only need to define them using arithmetical types.

Definition 4.16 (B^p, B^*)**.** For each bar recursor $B_{\sigma,\tau}$, we define the terms

$$B^p_{\sigma,\tau} \quad \text{as} \quad \lambda y, f, g, n, x. B_{\hat{\sigma},\hat{\tau}} y^m f^m g_f n x,$$
$$B^*_{\sigma,\tau} \quad \text{as} \quad \lambda y, f, g, n, x. (B_{\hat{\sigma},\hat{\tau}} y^m f^m g_f)^M n x,$$

where $y^m(x) = y(x^M)$, $f^m(n,x) = f(n, x^M)$, and $g_f(n,x,v) = \max\big(f(n,x^M), g(n,x^M,v)\big)$.

The theorem below is fundamental for proving that the bar recursors are strongly majorised functionals, that is, that they are part of $\mathscr{M}^{\omega,X}$.

Theorem 4.17. *Let $y \leq^* y^*$, $f \leq^* f^*$, and $g \leq^* g^*$ be of appropriate types for bar recursion, and $\hat{y} \leq^* y^*$, $\hat{f} \leq^* f^*$, and $\hat{g} \leq^* g^*$ be of the corresponding arithmetical types. Let r be an infinite sequence, and $\hat{r}$ and s be infinite sequences of the corresponding arithmetical type.*

For all $n \in \mathbb{N}$, it holds in $\mathscr{M}^{\omega,X}$ that, if $r(k) \leq^* s(k)$ and $\hat{r}(k) \leq^* s(k)$ for all $k < n$, then

$$B^p y^* f^* g^* n\hat{r} \leq^* B^p y^* f^* g^* ns$$
$$B^p \hat{y}\hat{f}\hat{g}n\hat{r} \leq^* B^p y^* f^* g^* ns$$
$$Byfgnr \leq^* B^p y^* f^* g^* ns.$$

Proof. The proof is based on the proofs of [20, Lemma 9.9] and [28, Lemma 17.83], and uses the same strategy as the proof of Theorem 3.12 — we use classical bar induction with the formula $P(n, s)$ being

$$\forall r^{0\to\sigma}\ \forall \hat{r}^{0\to\hat{\sigma}}\ \big(\forall k < n\,(rk, \hat{r}k \leq^* sk) \to Byfgnr, B^p\hat{y}\hat{f}\hat{g}n\hat{r}, B^p y^* f^* g^* n\hat{r} \leq^* B^p y^* f^* g^* ns\big).$$

The hypotheses are then proven using the generalised lemmas above. □

Corollary 4.18. *In $\mathscr{M}^{\omega,X}$ we have $B \leq^* B^*$.*

Proof. Just apply the argument for Corollary 3.13. □

Corollary 4.19. *$\mathscr{M}^{\omega,X}$ is a model of bar recursion, that is, a model of $\mathsf{PA}^{\omega,X} + \mathsf{BR}$.*

Proof. The bar recursors $B_{\sigma,\tau}$ are strongly majorised by $B^*_{\sigma,\tau}$. Hence $B_{\sigma,\tau}$ is in $\mathscr{M}^{\omega,X}$. □

4.5 Bar induction

Bar induction is to bar recursion as ordinary induction is to recursion — the former is the principle of proof corresponding to the latter. The version of bar induction we consider is *simplified monotone bar induction*, which is generalised to abstract types by letting σ be any extended type.

Definition 4.20 (Simplified monotone bar induction)**.** Given a formula $P(n^0, x^{0\to\sigma})$, the principle of *simplified monotone bar induction*, named BI^-, is

$$\text{Hyp1} \wedge \text{Hyp2} \wedge \text{Hyp3} \to P(0, 0^{0\to\sigma}),$$

where

- Hyp1 is $\forall x^{0\to\sigma}\ \exists n^0\ P(n, \overline{x|n})$ (well-foundedness)
- Hyp2 is $\forall n^0\ \forall x^{0\to\sigma}\ \forall i \leq n\ \big(P(i, \overline{x|i}) \to P(n, \overline{x|n})\big)$ (monotonicity)
- Hyp3 is $\forall n^0\ \forall x^{0\to\sigma}\ \big(\forall w^\sigma\ P(n+1, \overline{x|n * \langle w\rangle}) \to P(n, \overline{x|n})\big)$ (step)

When we restrict $P(n, x)$ to be an existential bounded formula, this principle is called *existential simplified monotone bar induction* and denoted by $\mathsf{BI}^-_\exists$.

By Theorem 3.20, we know that the theory $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{BR} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \Delta_{\mathscr{M}^\omega}$ proves $\mathsf{BI}^-_\exists$. Our goal is now to have the same result for $\mathsf{PA}^{\omega,X}_\trianglelefteq$ using a similar argument.

The first step is to prove Kreisel's trick for $\mathsf{PA}^{\omega,X}$, showing that the bar recursion axioms restrict the functionals Y of type $(0 \to \sigma) \to 0$ to those that yield a well-founded tree T_Y in the sense of §3.1.

Lemma 4.21 (Kreisel's trick extended)**.** *The theory* $\mathsf{PA}^{\omega,X} + \mathsf{BR}$ *proves*

$$\forall Y^{(0\to\sigma)\to 0}\, \forall x^{0\to\sigma}\, \exists i^0 \left(Y(\overline{x|i}) \le i\right),$$

for any extended type σ*.*

Proof. As definitions by bar recursion in $\mathsf{PA}^{\omega,X}$ allow the use of extended types, the proof used for Lemma 3.17 goes through without any changes. □

The second step is to have some additional axioms from $\mathscr{M}^{\omega,X}$, a model that allows definitions by bar recursion. One of these axioms is a monotonicity property for the bar recursor B^p.

Definition 4.22 ($\Delta_{\mathscr{M}^{\omega,X}}$)**.** We denote by $\Delta_{\mathscr{M}^{\omega,X}}$ the set of universal bounded sentences of the language of $\mathsf{PA}^{\omega,X}_{\trianglelefteq}$ whose flattening holds in $\mathscr{M}^{\omega,X}$.

Flattening a formula means replacing the intensional majorisability symbols $\trianglelefteq$ by the defined extensional symbol $\le^*$. The use of these universal bounded sentences is not by chance: with formulas of that form, we can use the soundness theorem to ensure that the theory $\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{BR} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \Delta_{\mathscr{M}^{\omega,X}}$ is still consistent.

Theorem 4.23 (Soundness for $\Delta_{\mathscr{M}^{\omega,X}}$)**.** *Let* $A(z)$ *be a formula of the language of* $\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{BR}$ *with free variables* z*. If*

$$\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{BR} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \Delta_{\mathscr{M}^{\omega,X}} \vdash A(z),$$

then there are closed monotone terms t*, effectively obtainable from a proof of* $A(z)$*, such that*

$$\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{BR} + \Delta_{\mathscr{M}^{\omega,X}} \vdash \widetilde{\forall} w\, \forall z \trianglelefteq w\, \widetilde{\forall} x\; A(z)_U(x, twx).$$

Proof. Apply the extended soundness theorem (Theorem 4.5) taking $\Delta = \mathsf{BR} + \Delta_{\mathscr{M}^{\omega,X}}$. □

By the theorem above, if the theory $\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{BR} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \Delta_{\mathscr{M}^{\omega,X}}$ was inconsistent, then $\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{BR} + \Delta_{\mathscr{M}^{\omega,X}}$ would also be inconsistent. But $\mathscr{M}^{\omega,X}$ is a model of $\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{BR} + \Delta_{\mathscr{M}^{\omega,X}}$, therefore the theory $\mathsf{PA}^{\omega,X}_{\trianglelefteq}$ with the characteristic principles and the axioms from $\mathscr{M}^{\omega,X}$ is consistent. Hence, it makes sense to prove the principle bar induction for this theory.

Theorem 4.24 (Bar induction)**.** *The theory* $\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{BR} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \Delta_{\mathscr{M}^{\omega,X}}$ *proves* $\mathsf{BI}^{-}_{\exists}$*.*

Proof. This proof is an adaptation of the proof of Theorem 3.20 to the setting with an abstract type.

Consider $P(n^0, r^{0\to\sigma})$ as the formula $\exists a^\tau\, Q(n, r, a)$, with Q a bounded formula. Without loss of generality, we can assume that τ is an arithmetical type and that $Q(n, r, a)$ is upward monotone in a in the sense that $a \trianglelefteq b$ and $Q(n, r, a)$ imply $Q(n, r, b)$. Indeed, let $P(n, r)$ be written in the form $\exists b^\rho\, P_{bd}(n, r, b)$, with P_{bd} a bounded formula. By MAJ, this formula is equivalent to $\exists a^{\hat\rho}\, \exists b \trianglelefteq a\; P_{bd}(n, r, b)$. Taking $Q(n, r, a)$ as $\exists b \trianglelefteq a\; P_{bd}(n, r, b)$, we get and upward monotone formula with $\tau = \hat\rho$ an arithmetical type.

From this point on the proof proceeds as that of Theorem 3.20. Assume the hypotheses of $\mathsf{BI}^{-}_{\exists}$:

- $\forall r^{0\to\sigma}\, \exists n^0\, \exists a^\tau\; Q(n, \overline{r|n}, a)$ (Hyp1)
- $\forall n^0\, \forall r^{0\to\sigma}\, \forall i \le n\, \left(\exists a^\tau\, Q(i, \overline{r|i}, a) \to \exists b^\tau\, Q(n, \overline{r|n}, b)\right)$ (Hyp2)

- $\forall n^0 \, \forall r^{0\to\sigma} \, \big(\forall w^\sigma \, \exists a^\tau \, Q(n+1, \overline{r|n * \langle w\rangle}, a) \to \exists b^\tau \, Q(n, \overline{r|n}, b)\big)$ (Hyp3)

By Proposition 4.4, we can use the principle of bounded choice bAC, since it can be proved by $\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{mAC} + \mathsf{bC}$. Using this principle on the hypotheses of bar induction and the monotonicity of Q, we get monotone functionals (of arithmetical type) Y, H, F, and G such that

- $\widetilde{\forall} s^{0\to\hat\sigma} \, \forall r \trianglelefteq s \, \exists n \leq Ys \;\; Q(n, \overline{r|n}, Hs)$ (Hyp1')
- $\forall n^0 \, \widetilde{\forall} s^{0\to\hat\sigma} \, \forall r \trianglelefteq s \, \widetilde{\forall} c^\tau \, \forall a \trianglelefteq c \, \big(\exists i \leq n \, Q(i, \overline{r|i}, a) \to Q(n, \overline{r|n}, F(n,s,c))\big)$ (Hyp2')
- $\forall n^0 \, \widetilde{\forall} s^{0\to\hat\sigma} \, \forall r \trianglelefteq s \, \widetilde{\forall} g^{\hat\sigma\to\tau} \, \forall f \trianglelefteq g \, \big(\widetilde{\forall} z^{\hat\sigma} \, \forall w \trianglelefteq z \; Q(n+1, \overline{r|n * \langle w\rangle}, fz) \to Q(n, \overline{r|n}, G(n,s,g))\big)$ (Hyp3')

Using the bar recursor B^p from Definition 4.16, we define

$$B(n,s) = B^p\big(Y, \lambda n, s.F(n,s,H(\overline{s|i_0})^M), G, n, s\big)$$
$$= \begin{cases} F\big(n, (\overline{s|n})^M, H(\overline{s|i_0})^M\big) & \text{if } \exists i \leq n \, \big(Y(\overline{s|i})^M \leq i\big) \\ \max\Big(F\big(n, (\overline{s|n})^M, H(\overline{s|i_0})^M\big), G\big(n, (\overline{s|n})^M, \lambda w.B(n, \overline{s|n * \langle w\rangle})\big)\Big) & \text{otherwise,} \end{cases}$$

where i_0 is the least $i \leq n$ such that $Y(\overline{s|i})^M \leq i$, if such i exists, and n otherwise.

Firstly, we claim that, for all n of type 0 and infinite sequences r and s such that $r(k) \trianglelefteq s(k)$ for all $k < n$, we have

$$\widetilde{\forall} z^{\hat\sigma} \; \forall w \trianglelefteq z \; Q(n+1, \overline{r|n * \langle w\rangle}, B(n+1, \overline{s|n * \langle z\rangle})) \to Q(n, \overline{r|n}, B(n, \overline{s|n})).$$

This is proven by considering the two cases of bar recursion

- $\exists i \leq n \, \big(Y(\overline{s|i})^M \leq i\big) \to Q(n, \overline{r|n}, B(n, \overline{s|n}))$ (1a)
- $\forall i \leq n \, \big(Y(\overline{s|i})^M > i\big) \to \big(\widetilde{\forall} z^\sigma \; \forall w \trianglelefteq z \; Q(n+1, \overline{r|n * \langle w\rangle}, B(n+1, \overline{s|n * \langle z\rangle})) \to Q(n, \overline{r|n}, B(n, \overline{s|n}))\big)$ (1b)

as in the proof of Theorem 3.20, but now using Lemma 4.13 rather than Lemma 3.8.

Applying the principle of bounded choice bAC to this claim provides us with a monotone functional $T \colon 0 \to (0 \to \hat\sigma) \to (0 \to \hat\sigma) \to \hat\sigma$ such that, for all n of type 0 and infinite sequences u and v, and for all $r \trianglelefteq u$ and $s \trianglelefteq v$, if r and s are such that $r(k) \trianglelefteq s(k)$ for all $k < n$, then we have

$$\widetilde{\forall} z^{\hat\sigma} \trianglelefteq T(n,u,v) \; \forall w \trianglelefteq z \; Q(n+1, \overline{r * \langle w\rangle}, B(n, \overline{s|n * \langle z\rangle})) \to Q(n, \overline{r|n}, B(n, \overline{s|n})).$$

Secondly, define a functional $x \colon 0 \to \hat\sigma$ by recursion with

$$x(0) = T(0, 0^{0\to\hat\sigma}, 0^{0\to\hat\sigma}) \quad \text{and} \quad x(n+1) = T(n, (\overline{x|n})^M, (\overline{x|n})^M).$$

By induction on n, we prove that, for all n of type 0, and for all infinite sequences r and s, if

$$\forall k < n \, (r(k) \trianglelefteq s(k) \trianglelefteq x(k)) \to Q(n, \overline{r|n}, B(n, \overline{s|n})),$$

then it follows that $Q(0, 0^{0\to\sigma}, B(0, 0^{0\to\hat\sigma}))$. The proof by induction is similar to the one in the proof of Theorem 3.20.

Lastly, use the extended version of Kreisel's trick to get n_0 such that $Y(\overline{x|n_0})^M \leq n_0$, that is, apply Lemma 4.21 to the functional $Y^m(x) = Y(x^M)$. Then, we use the second claim to conclude the proof. Let r and s be arbitrary infinite sequences such that $r(k) \trianglelefteq s(k) \trianglelefteq x(k)$ for all $k < n_0$. By Lemma 4.13, we have $(\overline{s|n_0})^M \trianglelefteq (\overline{x|n_0})^M$, and, therefore, it follows that $Y(\overline{s|n_0})^M \leq Y(\overline{x|n_0})^M \leq n_0$. Using (1a), it follows that $Q(n_0, \overline{r|n_0}, B(n_0, \overline{s|n_0}))$, proving

$$\forall k < n_0 \, (r(k) \trianglelefteq s(k) \trianglelefteq x(k)) \to Q(n, \overline{r|n_0}, B(n_0, \overline{s|n_0})),$$

and using the second claim, we conclude $Q(0, 0^{0\to\sigma}, B(0, 0^{0\to\hat{\sigma}}))$. Hence we have proven $\exists a^\tau \, Q(0, 0^{0\to\sigma}, a)$, that is, $P(0, 0^{0\to\sigma})$, as we wanted. ☐

In the previous proof, we have used some sentences from $\Delta_{\mathscr{M}^\omega, x}$ without explicit reference. We now highlight them.

- The main use of $\Delta_{\mathscr{M}^\omega, x}$ is through Theorem 4.24, via the sentence

$$\widetilde{\forall} y, f, g \; \forall n^0 \; \forall r^{0\to\sigma} \; \forall s^{0\to\hat{\sigma}} \; \big(\forall k < n \, (rk \trianglelefteq sk) \to B^p yfgnr \trianglelefteq B^p yfgns\big).$$

- Lemma 4.13 is used in the form

$$\forall n^0 \; \forall r^{0\to\sigma} \; \forall s^{0\to\hat{\sigma}} \; \big(\forall k < n \, (rk \trianglelefteq sk) \to \overline{r|n} \trianglelefteq (\overline{s|n})^M\big),$$

  when σ is any extended type, and

$$\forall n^0 \; \forall r^{0\to\sigma} \; \forall s^{0\to\hat{\sigma}} \; \big(\forall k < n \, (rk \trianglelefteq sk) \to (\overline{r|n})^M \trianglelefteq (\overline{s|n})^M\big),$$

  when σ is of the form $\sigma_1 \to \cdots \to \sigma_k \to 0$.

- Bounded extensionality from $\Delta_{\mathscr{M}^\omega, x}$ is used for substitutions of the form

$$\forall n^0 \; \forall r^{0\to\sigma}, s^{0\to\sigma} \; \big(\forall k < n \, (rk = sk) \wedge A_{bd}[\overline{r|n}/z] \to A_{bd}[\overline{s|n}/z]\big),$$

  for bounded formulas $A_{bd}(z)$.

4.6 Numerical comprehension

The path to numerical comprehension is now the same we trod in §3.4: having added bar recursion and bar induction to the theory of finite-type arithmetic, we also get the principle of dependent choices.

Definition 4.25 (DC^ω)**.** The principle of *dependent choices*, denoted by DC^ω, is

$$\forall x^\sigma \, \exists y^\sigma \, A(x, y) \to \forall w^\sigma \, \exists f^{0\to\sigma} \, \big(f(0) = w \wedge \forall n^0 \, A(f(n), f(n+1))\big),$$

where $A(x, y)$ can be any formula where f does not occur free, and σ any tuple of extended types. When we restrict $A(x, y)$ to be a universal bounded formula, this principle is denoted by $\mathsf{DC}^\omega_\forall$.

In a sense, the principle of dependent choices DC^ω is the contrapositive of bar induction BI^-, as seen in the proof of Proposition 3.24.1. From dependent choices, we derive the principle of numerical choice.

Definition 4.26 ($\mathsf{AC}^{0,\omega}$)**.** The principle of *numerical choice*, denoted by $\mathsf{AC}^{0,\omega}$, is

$$\forall n^0 \, \exists x^\sigma \, A(n,x) \to \exists f^{0\to\sigma} \, \forall n^0 \, A(n, f(n)),$$

where $A(n,x)$ can be any formula where f does not occur free, and σ any tuple of extended types. When we restrict σ to be type 0, this principle is denoted by $\mathsf{AC}^{0,0}$.

By a simple application of the principle of numerical choice, we conclude the principle of numerical comprehension.

Definition 4.27 (CA^0)**.** The principle of *numerical comprehension*, denoted by CA^0, is

$$\exists f^{0\to 0} \, \forall n^0 \, (f(n) = 0 \leftrightarrow A(n)),$$

where $A(n)$ can be any formula where f does not occur free.

In short, we have

Proposition 4.28.

1. $\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{BI}^-_\exists \vdash \mathsf{DC}^\omega_\forall$,
2. $\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \mathsf{DC}^\omega_\forall \vdash \mathsf{DC}^\omega$,
3. $\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{DC}^\omega \vdash \mathsf{AC}^{0,\omega}$,
4. $\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{AC}^{0,0} \vdash \mathsf{CA}^0$.

Proof. Use the same arguments as for Proposition 3.24, but now using the extended types and the extension of bar induction to those types. □

Corollary 4.29. *The theory* $\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{BR} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \Delta_{\mathscr{M}^{\omega,X}}$ *proves* CA^0.

Space completion In an application of the principles deduced above, we prove that the bounded functional interpretation automatically completes metric spaces. In particular, we show that, in the presence of bar recursion, the principle of bounded collection bC introduces limits for all Cauchy sequences.

A sequence $x\colon 0 \to X$ is a *Cauchy sequence* if it satisfies

$$\forall k^0 \; \exists n^0 \; \forall i,j \geq n \; \left(d(xi,xj) \leq_{\mathbb{R}} \frac{1}{k+1} \right),$$

and a *rate of convergence* for a Cauchy sequence is a function $f\colon 0 \to 0$ such that

$$\forall k^0 \; \forall i,j \geq fk \; \left(d(xi,xj) \leq_{\mathbb{R}} \frac{1}{k+1} \right).$$

The existence of a rate of convergence implies that a sequence is a Cauchy sequence. In the presence of numerical choice $\mathsf{AC}^{0,0}$, the converse holds — i.e., every Cauchy sequence has a rate of convergence.

Proposition 4.30. *Consider a Cauchy sequence* $x\colon 0 \to X$*. The theory* $\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{BR} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \Delta_{\mathscr{M}^{\omega,X}}$ *proves that* x *has a limit, that is,*

$$\exists z^X \; \forall k^0 \; \forall i \geq fk \; \left(d(xi,z) \leq_{\mathbb{R}} \frac{1}{k+1} \right).$$

Proof. This proof adapts the proof of [17, Proposition 1] to a general a metric space.

As x is a Cauchy sequence, by applying the principle of numerical choice $\mathsf{AC}^{0,0}$, we get a rate of convergence $f\colon 0 \to 0$ for x. Without loss of generality, we can assume that f is monotone.

By MAJ, there is a bound m such that $d(x(f0), 0^X) \leq m$. Aiming to apply bC, we claim that

$$\forall a^0, b^0\ \exists z \trianglelefteq m+1\ \forall k \leq a\ \forall i \leq b\ \left(i \geq fk \to d(xi, z) \leq_{\mathbb{R}} \frac{1}{k+1}\right).$$

(The bound $i \leq b$ is not necessary for the proof, but we write the claim in this form to apply bC)
To prove the claim, consider a and b of type 0 and take $z = x(fa)$. As f is monotone, we get $f0 \leq fa$. Hence, by hypothesis, it follows that $d(z, x(f0)) = d(x(fa), x(f0)) \leq_{\mathbb{R}} 1$. Therefore

$$d(z, 0^X) \leq_{\mathbb{R}} d(z, x(f0)) + d(x(f0), 0^X) \leq_{\mathbb{R}} m+1,$$

that is, $z \trianglelefteq m+1$. Now let $k \leq a$ and $i \leq b$ with $i \geq fk$. By monotonicity of f, we get $fa \geq fk$. Thus, by hypothesis, we have $d(xi, x(fa)) \leq_{\mathbb{R}} \frac{1}{k+1}$, as we wanted. Now apply bC to the claim to get the result. □

Chapter 5

Epilogue

We have studied two extensions to the bounded functional interpretation, in the context of classical logic, that allow the interpretation of stronger and richer theories. These extensions had been previously studied in separation, for intuitionistic logic, in [9].

The first extension extended the bounded functional interpretation with bar recursive functionals in a similar way to how Spector [39] extends Gödel's dialectica interpretation. This extension allowed the interpretation of theories that prove the principle of dependent choices and the principle of numerical comprehension. In particular, we have checked that, for the interpretation of the principle of arithmetical comprehension, we only required the bar recursor $B_{0,0}$. These results have been applied for the interpretation of subsystems of second-order arithmetic, namely RCA, ACA, and PA_2 itself.

The second extension involved the addition of an abstract type — an additional ground type representing some mathematical structure, such as a metric space or a ring — using a method inspired by Kohlenbach's extensions [27, 20] to the monotone functional interpretation. With an abstract type, the bounded functional interpretation can interpret theories about spaces that do not have to be encoded.

We have checked in detail that the bounded functional interpretation can comport both extensions at the same time. Indeed, we have successfully adapted, in the presence of an abstract type, the arguments that prove the principle of dependent choices from the proof principle of bar induction — the correspondent of bar recursion. In an application, we have shown that the extended bounded functional interpretation automatically completes metric spaces.

Future work

The logical sequel to this work is the application of the bounded functional interpretation, now enriched with bar recursion and an abstract type, to studies in proof mining. The first step in this direction is taken in [35], the first use of the bounded functional interpretation for the proof mining of concrete mathematical results in analysis. However, the theorems analysed there do not require the use of bar recursion, even though they make use of an abstract type. The added strength of bar recursion may enable the employment of proof mining techniques to a wider range of theorems.

Speaking of a wider range, a parallel objective is the application of methods of proof mining to more branches of mathematics. The pioneering work of Kohlenbach [26] bridges between logic and analysis — the core results are from logic, but their deployment leads to new purely analytical results. Since then, proof mining has been mostly focused on theorems from analysis. At any rate, there is no impediment to the adoption of proof mining methods in other areas, as is evidenced by Ferreira in [15], where the bounded functional interpretation is used to obtain proof mining results from a theorem in algebra.

More concrete suggestions for future work include a possible embellishment to our developments about bar recursion, replacing the use of $\Delta_{\mathscr{M}^\omega}$ — the set of universal bounded sentences whose flattening holds in $\mathscr{M}^\omega$ — with a more restricted set of axioms, akin to the list of sentences after the proof of Theorem 3.20. A worthwhile further question is whether all of these sentences would have to be added as axioms, or if the theories in question can already prove some of them.

Another concrete question concerns the computation of bounds for the interpretations of choice and comprehension principles proven using bar induction. For example, by Proposition 4.28.3, the theory $\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{BR} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \Delta_{\mathscr{M}^{\omega,X}}$ proves the principle of numerical choice $\mathsf{AC}^{0,0}$. The soundness theorem for bar recursion, Theorem 4.23, guarantees the existence of bounding terms t including the bar recursors such that $\mathsf{PA}^{\omega}_{\trianglelefteq} + \mathsf{BR} + \Delta_{\mathscr{M}^\omega}$ proves $\tilde{\forall} x\ \mathsf{AC}^{0,0}_U(x, tx)$, where $\tilde{\forall} x\, \tilde{\exists} y\ \mathsf{AC}^{0,0}_U(x, y)$ is the bounded functional interpretation of $\mathsf{AC}^{0,0}$. Even though the interpretation of $\mathsf{AC}^{0,0}$ is admittedly a bit opaque, it would be enlightening to see the bar recursors occurring in the bounding terms.

Bibliography

[1] M. F. Atiyah and I. G. Macdonald. *Introduction to Commutative Algebra*. Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1969, pp. ix+128.

[2] B. van den Berg. 'A note on equality in finite-type arithmetic'. In: *MLQ Math. Log. Q.* 63.3-4 (2017), pp. 282–288.

[3] M. Bezem. 'Strongly majorizable functionals of finite type: a model for barrecursion containing discontinuous functionals'. In: *J. Symbolic Logic* 50.3 (1985), pp. 652–660.

[4] N. Bourbaki. *Algebra. I. Chapters 1–3.* Elements of Mathematics (Berlin). Translated from the French, Reprint of the 1974 edition. Springer-Verlag, Berlin, 1989, pp. xxiv+709.

[5] N. Bourbaki. *General Topology. Chapters 5–10.* Elements of Mathematics (Berlin). Translated from the French, Reprint of the 1989 English translation. Springer-Verlag, Berlin, 1998, pp. iv+363.

[6] L. E. J. Brouwer. 'Über Definitionsbereiche von Funktionen'. In: *Math. Ann.* 97.1 (1927), pp. 60–75.

[7] F. E. Browder. 'Convergence of approximants to fixed points of nonexpansive nonlinear mappings in Banach spaces'. In: *Arch. Rational Mech. Anal.* 24 (1967), pp. 82–90.

[8] H. B. Curry. 'Grundlagen der Kombinatorischen Logik'. In: *Amer. J. Math.* 52.4 (1930), pp. 789–834.

[9] P. Engrácia. 'Proof-theoretical studies on the bounded functional interpretation'. PhD thesis. 2009.

[10] P. Engrácia. 'The bounded functional interpretation of bar induction'. In: *Ann. Pure Appl. Logic* 163.9 (2012), pp. 1183–1195.

[11] P. Engrácia and F. Ferreira. 'Bounded functional interpretation with an abstract type'. In: *Contemporary Logic and Computing.* Coll. Publ., London, 2020, pp. 87–112.

[12] F. Ferreira. 'Injecting uniformities into Peano arithmetic'. In: *Ann. Pure Appl. Logic* 157.2-3 (2009), pp. 122–129.

[13] F. Ferreira. 'Proof interpretations and majorizability'. In: *Logic Colloquium 2007.* Vol. 35. Lect. Notes Log. Assoc. Symbol. Logic, La Jolla, CA, 2010, pp. 32–81.

[14] F. Ferreira. 'Spector's proof of the consistency of analysis'. In: *Gentzen's Centenary.* Springer, Cham, 2015, pp. 279–300.

[15] F. Ferreira. 'Bounds for indexes of nilpotency in commutative ring theory: a proof mining approach'. In: *Bull. Symb. Log.* 26.3-4 (2020), pp. 257–267.

[16] F. Ferreira. 'The FAN principle and weak König's lemma in herbrandized second-order arithmetic'. In: *Ann. Pure Appl. Logic* 171.9 (2020), pp. 102843, 21.

[17] F. Ferreira. 'The abstract type of the real numbers'. In: *Arch. Math. Logic* 60.7 (2021), pp. 1005–1017.

[18] F. Ferreira, L. Leuştean and P. Pinto. 'On the removal of weak compactness arguments in proof mining'. In: *Adv. Math.* 354 (2019), pp. 106728, 55.

[19] F. Ferreira and P. Oliva. 'Bounded functional interpretation'. In: *Ann. Pure Appl. Logic* 135.1-3 (2005), pp. 73–112.

[20] P. Gerhardy and U. Kohlenbach. 'General logical metatheorems for functional analysis'. In: *Trans. Amer. Math. Soc.* 360.5 (2008), pp. 2615–2660.

[21] K. Gödel. 'Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes'. In: *Dialectica* 12 (1958), pp. 280–287.

[22] W. A. Howard. 'Functional interpretation of bar induction by bar recursion'. In: *Compositio Math.* 20 (1968), pp. 107–124.

[23] W. A. Howard. 'Appendix: Hereditarily majorizable functionals of finite type'. In: *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis.* 1973, 454–461. Lecture Notes in Math., Vol. 344.

[24] W. A. Howard and G. Kreisel. 'Transfinite induction and bar induction of types zero and one, and the role of continuity in intuitionistic analysis'. In: *J. Symbolic Logic* 31.3 (1966), pp. 325–358.

[25] S. C. Kleene. 'Recursive functions and intuitionistic mathematics'. In: *Proceedings of the International Congress of Mathematicians, Cambridge, Mass., 1950, vol. 1.* Amer. Math. Soc., Providence, R. I., 1952, pp. 679–685.

[26] U. Kohlenbach. 'Analysing proofs in analysis'. In: *Logic: from Foundations to Applications (Staffordshire, 1993).* Oxford Sci. Publ. Oxford Univ. Press, New York, 1996, pp. 225–260.

[27] U. Kohlenbach. 'Some logical metatheorems with applications in functional analysis'. In: *Trans. Amer. Math. Soc.* 357.1 (2005), pp. 89–128.

[28] U. Kohlenbach. *Applied Proof Theory: Proof Interpretations and Their Use in Mathematics.* Springer Monographs in Mathematics. Springer-Verlag, Berlin, 2008, pp. xx+532.

[29] G. Kreisel. 'On the interpretation of non-finitist proofs. I'. In: *J. Symbolic Logic* 16.4 (1951), pp. 241–267.

[30] G. Kreisel. 'On the interpretation of non-finitist proofs. II. Interpretation of number theory. Applications'. In: *J. Symbolic Logic* 17.1 (1952), pp. 43–58.

[31] G. Kreisel. 'Mathematical significance of consistency proofs'. In: *J. Symbolic Logic* 23.2 (1958), pp. 155–182.

[32] G. Kreisel. 'Interpretation of analysis by means of constructive functionals of finite types'. In: *Constructivity in Mathematics: Proceedings of the Colloquium held at Amsterdam, 1957 (edited by A. Heyting)*. Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Co., Amsterdam, 1959, pp. 101–128.

[33] S. Kuroda. 'Intuitionistische Untersuchungen der formalistischen Logik'. In: *Nagoya Math. J.* 2 (1951), pp. 35–47.

[34] H. Luckhardt. *Extensional Gödel Functional Interpretation. A Consistency Proof of Classical Analysis*. Lecture Notes in Mathematics, Vol. 306. Springer-Verlag, Berlin-New York, 1973, pp. vi+161.

[35] P. Pinto. 'Proof mining with the bounded functional interpretation'. PhD thesis. 2019.

[36] B. Scarpellini. 'A model for barrecursion of higher types'. In: *Compositio Math.* 23 (1971), pp. 123–153.

[37] J. R. Shoenfield. *Mathematical Logic*. Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1967, pp. viii+344.

[38] S. G. Simpson. *Subsystems of second order arithmetic*. Second. Perspectives in Logic. Cambridge University Press, Cambridge; Association for Symbolic Logic, Poughkeepsie, NY, 2009, pp. xvi+444.

[39] C. Spector. 'Provably recursive functionals of analysis: a consistency proof of analysis by an extension of principles formulated in current intuitionistic mathematics'. In: *Proc. Sympos. Pure Math., Vol. V.* American Mathematical Society, Providence, R.I., 1962, pp. 1–27.

[40] W. W. Tait. 'Normal form theorem for bar recursive functions of finite type'. In: *Proceedings of the Second Scandinavian Logic Symposium (Univ. Oslo, Oslo, 1970)*. 1971, 353–367. Studies in Logic and the Foundations of Mathematics, Vol. 63.

[41] A. S. Troelstra, ed. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*. Lecture Notes in Mathematics, Vol. 344. Springer-Verlag, Berlin-New York, 1973, pp. xvii+485.

[42] A. S. Troelstra. 'Introductory note to 1958 and 1972'. In: *Kurt Gödel Collected Works*. Ed. by S. Feferman et al. Vol. 2. Oxford Univ. Press, New York, 1990, pp. 217–241.