Bar recursion with abstract types

a study on the bounded functional interpretation

João Rafael Oliveira Mira da Silva

October 2021

Abstract

We study two extensions to the bounded functional interpretation of Ferreira and Oliva, in the context of classical logic. The first extension introduces definitions by bar recursion, a form of transfinite recursion in well-founded trees. The second extension concerns the addition of an abstract type, that is, the inclusion of an additional ground type which can represent a ring, or a metric space, for example.

Our goal is to study the simultaneous extension of the bounded functional interpretation with bar recursion and an abstract type. We deduce, in the presence of an abstract type, the principle of dependent choices from bar induction, the proof principle corresponding to bar recursion.

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

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.

Proof interpretations were introduced in the context of relative consistency proofs. Nowadays, the more interesting facet of these interpretations is that they 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, whose main question is posed by Georg Kreisel:

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

The main technique at work in proof mining is the *monotone functional interpretation* of Kohlenbach [5], an interpretation that allows the extraction of bounds for witnessing terms. In exchange for only extracting bounding terms, the monotone interpretation can deal with some non-constructive principles, including, for instance, weak König's lemma — the claim that every infinite binary tree has an infinite path.

A different interpretation used for proof mining is the *bounded functional interpretation* of Fernando Ferreira and Paulo Oliva [4]. 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.

In this work, we present a version of the bounded functional interpretation directly for classical logic [3]. We follow two lines of inquiry,

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

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. These extensions to the bounded functional interpretation have been studied individually in the PhD thesis of Patrícia Engrácia [1], 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.

The former item settles a question posed by Engrácia and Ferreira in [2], and is relevant for the proof mining programme in the context of the bounded functional interpretation.

2 Bounded functional interpretation

In this section, we present the theory of finite-type arithmetic, a version of this theory with intensional majorisability, and the bounded functional interpretation — an interpretation of finite-type arithmetic with intensional majorisability into itself.

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* — and 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.

Language The language of finite-type arithmetic has variables $x^\tau, y^\tau, z^\tau, \ldots$ of each type τ. There are the logical constants Π and Σ, called the *combinators*. The combinator Π represents the function $(x, y) \mapsto x$ and Σ represents the function $(x, y, z) \mapsto x(z, y(z))$. There are also the arithmetical constants: 0 represents the first natural number, S represents the successor function $n \mapsto n + 1$, and the constants

R_τ, named *recursors*, are used to define functions by recursion, they represent the function g such that $g(0, x, f) = x$, and $g(n+1, x, f) = f(g(n, x, f), n)$.

There is a unique binary predicate symbol $=_0$ between objects of type 0.

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, and 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.

Terms are usually written with the type superscript notation t^τ, meaning that t is a term of type τ. 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 using the connectives $\neg$ and $\vee$ and the quantifiers $\forall x^\tau$, for each type τ. The other connectives and the existential quantifier are defined by abbreviation.

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 $x =_0 x$ and $x =_0 y \wedge A[x/w] \to A[y/w]$, where $A(w)$ is an atomic formula,
- successor axioms $Sx \neq_0 0$ and $Sx =_0 Sy \to x =_0 y$,
- combinator axioms $A[\Pi xy/w] \leftrightarrow A[x/w]$ and $A[\Sigma xyz/w] \leftrightarrow A[xz(yz)/w]$, where $A(w)$ is an atomic formula,
- recursor axioms $A[R(0, y, z)/w] \leftrightarrow A[y/w]$ and $A[R(Sx, y, z)/w] \leftrightarrow A[z(R(x, y, z), x)/w]$, 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.

In addition to the primitive equality $=_0$ between objects of type 0, we define equality for higher types $t =_{\sigma\to\tau} s$ as $\forall x^\sigma\, (tx =_\tau sx)$, assuming that x is not a variable of the terms t and s.

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. The usual order relation $\leq$ and the maximum operator max can be defined primitive recursively.

Standard model The *full set-theoretic model*, denoted by $\mathscr{S}^\omega$, is the standard model for PA^ω. The type structure $\mathscr{S}^\omega$ is defined recursively as $S_0 = \mathbb{N}$, and $S_{\sigma\to\tau} = S_\tau^{S_\sigma}$, for all types σ and τ.

Majorisability 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 $M_0 = \mathbb{N}$; and $x \leq^*_{\sigma\to\tau} y$ 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.

The relation $\leq^*$ is not reflexive, but if we have $x \leq^* y$, then $y \leq^* y$. Furthermore, the strong majorisability relation 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. By finding majorants for the constant symbols of the language, we conclude that $\mathscr{M}^\omega$ is a model of PA^ω.

2.2 Intensional majorisability

In PA^ω, the *extensional* version of strong majorisability $\leq^*$ is obtained by defining the symbols $\leq^*$ according to the definition of strong majorisability above. For the *intensional* version of strong majorisability $\trianglelefteq$ — a version partially governed by a rule — we add binary relation symbols $\trianglelefteq_\tau$ to our language and *bounded quantifiers* $\forall x \trianglelefteq_\tau t$ where the variable x does not occur in the term t. The bounded quantifier $\exists x \trianglelefteq_\tau t$ is defined by abbreviation. 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.*

The *theory of Peano arithmetic in all finite types with intensional majorisability*, denoted by $\mathsf{PA}^\omega_\trianglelefteq$, is the extension of PA^ω with the universal closures of:

- bounded quantifier axioms $\forall x \trianglelefteq_\tau t\, A \leftrightarrow \forall x\, (x \trianglelefteq_\tau t \to A)$, where A is an arbitrary formula, the variable x does not occur in the term t, and τ is any type,
- strong majorisability axioms $x \trianglelefteq_0 y \leftrightarrow x \leq_0 y$ and, for all types σ and τ,

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

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.

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^*.

2.3 Bounded functional interpretation

The bounded functional interpretation is a proof interpretation of $\mathsf{PA}^\omega_\trianglelefteq$ in itself.

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

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

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 $\tilde{\forall} f\ \tilde{\exists} x\ \exists x' \trianglelefteq x\ \neg A_U(x', fx')$,
- $(A \vee B)^U$ 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.

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 2.4):

- the *monotone bounded choice* principle, denoted by mAC, is
  $\tilde{\forall} x\ \tilde{\exists} y\ A_{bd}(x, y) \to \tilde{\exists} f\ \tilde{\forall} x\ \tilde{\exists} y \trianglelefteq fx\ A_{bd}(x, y)$
  with $A_{bd}(x, y)$ a bounded formula where f does not occur free, and x and y are tuples of variables,
- the *bounded collection* principle, denoted by bC, is
  $\forall x \trianglelefteq a\ \exists y\ A_{bd}(x, y) \to \tilde{\exists} b\ \forall x \trianglelefteq a\ \exists y \trianglelefteq b\ A_{bd}(x, y)$
  with $A_{bd}(x, y)$ a bounded formula where b does not occur free, and x and y are tuples of variables,
- the *majorisability* principle, denoted by MAJ, is $\forall x\, \exists y\, (x \trianglelefteq y)$.

The following results are the main theorems of the bounded functional interpretation.

Theorem 2.2 (Soundness)**.** *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*

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

Corollary 2.3 (Extraction)**.** *Let $A_{bd}(x, y)$ be a bounded formula of $\mathsf{PA}^\omega_\trianglelefteq$ 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*

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

Theorem 2.4 (Characterisation)**.** *Let A be a formula of the language of $\mathsf{PA}^\omega_\trianglelefteq$. We have*

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

3 Bar recursion

In this section, we present an extension of the bounded functional interpretation with bar recursion and show that this extension enables the interpretation of proofs that use numerical comprehension.

3.1 Bar recursion

We begin with an informal discussion of bar recursion. 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^*. We consider 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.

Definition 3.1 (Bar recursion)**.** To define a function $S^* \to T$ by bar recursion, we require functions $Y\colon S^{\mathbb{N}} \to \mathbb{N}$, $F\colon S^* \to T$, and $G\colon S^* \times T^S \to T$. 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 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 only on the condition function Y. It can be proven that, if $Y \in M_{(0\to\sigma)\to 0}$, then the function $B(s)$ is a total function.

Bar recursion in finite-type arithmetic Bar recursion is defined using finite sequences, which are not directly available in finite-type Peano arithmetic. A finite sequence with elements of type σ is represented by a pair of objects: one of type 0 — its length — and the other of type $0 \to \sigma$ — an infinite sequence to be truncated. The *canonical representative* for a finite sequence is the pair $n, \overline{x|n}$ whose infinite sequence is extended with zeros.

To add bar recursion to PA^ω, we include additional constants — the *bar recursors* $B_{\sigma,\tau}$ — and the bar recursor axioms BR

$$\exists i \leq n \left(y(\overline{x|i}) \leq i\right) \to \left(A[B(y,f,g,n,x)/z] \leftrightarrow A[f(n,\overline{x|n})/z]\right)$$
$$\forall i \leq n \left(y(\overline{x|i}) > i\right) \to \left(A[B(y,f,g,n,x)/z] \leftrightarrow A[g(n,\overline{x|n}, \lambda w.B(y,f,g,Sn,\overline{x|n * \langle w\rangle}))/z]\right)$$

where $A(z)$ is an atomic formula.

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. 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.

3.2 Bar induction

Bar induction is the principle of proof corresponding to bar recursion, in the same manner as proofs by ordinary induction correspond to definitions by ordinary recursion. The version of bar induction we use is the following.

Definition 3.2 (Simplified monotone bar induction)**.** Given a formula $P(n^0, x^{0\to\sigma})$, the principle of *simplified monotone bar induction*, named BI^-, is the formula $\mathrm{Hyp1} \wedge \mathrm{Hyp2} \wedge \mathrm{Hyp3} \to P(0, 0^{0\to\sigma})$, where

- Hyp1 is $\forall x^{0\to\sigma}\ \exists n^0\ P(n, \overline{x|n})$ (well-foundedness)
- Hyp2 is $\forall n^0\ \forall x^{0\to\sigma}\ \forall i \le n\ \big(P(i, \overline{x|i}) \to P(n, \overline{x|n})\big)$ (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 $\mathsf{BI}^-_\exists$.

To prove bar induction, 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 $\mathscr{M}^\omega$.

Theorem 3.3 (Bar induction)**.** *The theory* $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{BR} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \Delta_{\mathscr{M}^\omega}$ *proves* $\mathsf{BI}^-_\exists$.

3.3 Numerical comprehension

The addition of bar recursion significantly strengthens the theory of finite-type arithmetic. In fact, this extension can be used to interpret proofs that use a number of choice and comprehension principles:

- the principle of *numerical comprehension* CA^0
  $\exists f^{0\to 0}\, \forall n^0\, (f(n) = 0 \leftrightarrow A(n))$
  where $A(n)$ can be any formula where f does not occur free,
- the principle of *dependent choices* DC^ω
  $\forall x^\sigma\, \exists y^\sigma\, A(x, y) \to \forall w^\sigma\, \exists f^{0\to\sigma}\, \big(f(0) = w \wedge \forall n^0\, A(f(n), f(n+1))\big)$
  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 $\mathsf{DC}^\omega_\forall$,
- the principle of *numerical choice* $\mathsf{AC}^{0,\omega}$
  $\forall n^0\, \exists x^\sigma\, A(n, x) \to \exists f^{0\to\sigma}\, \forall n^0\, A(n, f(n))$
  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 3.4.

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. $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{AC}^{0,0} \vdash \mathsf{CA}^0$.

Corollary 3.5. *The theory* $\mathsf{PA}^\omega_\trianglelefteq + \mathsf{BR} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \Delta_{\mathscr{M}^\omega}$ *proves* CA^0.

4 Abstract type

In this section, we develop an extension of the bounded functional interpretation simultaneously with bar recursion and with an abstract type that can represent a ring, or a metric space, for example. By using an abstract type, we are no longer restricted to working with spaces that have to be encoded.

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.

Types The *extended finite types* are defined recursively as follows: 0 and X are extended finite types — the *extended ground types*, and if σ and τ are extended finite types, then $\sigma \to \tau$ is an extended finite type — an *extended higher type*. The finite types using only ground type 0 are called *arithmetical types*.

Language The language of finite-type arithmetic is extended with a constant 0_X of type X and other new constants depending on the structure X at hand. For instance, if X is a ring, then one of these constants could be $+$ of type $X \to X \to X$ that represents the additive operation.

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 X: $x =_X x$ and $x =_X y \wedge A[x/w] \to A[y/w]$, where $A(w)$ is an atomic formula,
- universal axioms related to the structure X.

When X is a ring, we would have a distributivity axiom $(x + y) + z = x + (y + z)$.

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

Majorisability The *extended strong majorisability* relation 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 $M_0 = \mathbb{N}$; the relation $x \leq^*_X n$ 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 $M_X = X$; and $x \leq^*_{\sigma\to\tau} y$ 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 extended version coincides with the original version of strong majorisability 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: reflexivity for strong majorants and transitivity.

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.

Intensional majorisability In $\mathsf{PA}^{\omega,X}$, we can also define an *extensional* version of $\leq^*$ as above, assuming that $x \leq^*_X n$ is defined by a universal formula $B_X(n,x)$. The *intensional* version $\trianglelefteq$ requires the addition of symbols $\trianglelefteq_\tau$ and the bounded quantifiers as before.

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 the universal closures of the formulas:

- bounded quantifier axioms $\forall x \trianglelefteq_\tau t\, A \leftrightarrow \forall x\,(x \trianglelefteq_\tau t \to A)$, 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 $n \trianglelefteq_0 m \leftrightarrow n \leq_0 m$; $x \trianglelefteq_X n \to B_X(x,n)$; and, for all extended types σ and τ,

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

- universal bounded axioms related to the structure X,

and the extended strong majorisability rules

$$\frac{A_{bd} \to B_X(x,n)}{A_{bd} \to x \trianglelefteq_X n} \qquad \text{and} \qquad \frac{A_{bd} \wedge u \trianglelefteq_\sigma v \wedge u' \trianglelefteq_{\hat\sigma} v' \to tu \trianglelefteq_\tau sv \wedge su' \trianglelefteq_{\hat\tau} sv'}{A_{bd} \to t \trianglelefteq_{\sigma\to\tau} s}$$

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.

4.2 Extended bounded functional interpretation

The bounded functional interpretation as in Definition 2.1 also provides an interpretation of $\mathsf{PA}^{\omega,X}_{\trianglelefteq}$ into itself, with the characteristic principles mAC, bC, and MAJ generalised to the extended types.

Theorem 4.1 (Soundness)**.** *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*

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

Theorem 4.2 (Characterisation)**.** *Let A be a formula of the language of $\mathsf{PA}^{\omega,X}_{\trianglelefteq}$. We have*

$$\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} \vdash A \leftrightarrow A^U.$$

4.3 From bar recursion to numerical comprehension

The principle of definition by bar recursion can be extended to abstract types simply by allowing bar recursive definitions involving the extended types. As before, the extended model of strongly majorisable functionals is a model for bar recursion — that is, a model for the extended bar recursor axioms BR.

The principle of proof by bar induction BI^- can be similarly extended. To prove this extension of bar induction, we use sentences from the set $\Delta_{\mathscr{M}^{\omega,X}}$ of formulas whose flattening is true in $\mathscr{M}^{\omega,X}$.

Theorem 4.3 (Bar induction)**.** *The theory* $\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{BR} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \Delta_{\mathscr{M}^{\omega,X}}$ *proves* $\mathsf{BI}^-_{\exists}$*.*

Furthermore, using this extended bar induction, we can prove the same choice and comprehension principles, where the type σ can be any extended type.

Proposition 4.4.

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. $\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{AC}^{0,0} \vdash \mathsf{CA}^{0}$.

Corollary 4.5. *The theory* $\mathsf{PA}^{\omega,X}_{\trianglelefteq} + \mathsf{BR} + \mathsf{mAC} + \mathsf{bC} + \mathsf{MAJ} + \Delta_{\mathscr{M}^{\omega,X}}$ *proves* CA^0*.*

With a suitable axiomatisation of metric spaces, the principle of numerical choice $\mathsf{AC}^{0,0}$ can be used, in conjunction with the principle of bounded collection bC, to prove that the bounded functional interpretation automatically completes metric spaces.

Proposition 4.6. *Consider a Cauchy sequence* $x\colon 0 \to X$*, that is, a sequence* x *such that*

$$\forall k^0 \; \exists n^0 \; \forall i, j \geq n \; \left(d(xi, xj) \leq \frac{1}{k+1} \right).$$

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,*

$$\exists z^X \; \forall k^0 \; \forall i \geq fk \; \left(d(xi, z) \leq \frac{1}{k+1} \right).$$

References

[1] P. Engrácia. 'Proof-theoretical studies on the bounded functional interpretation'. PhD thesis. 2009.

[2] P. Engrácia and F. Ferreira. 'Bounded functional interpretation with an abstract type'. In: *Contemporary Logic and Computing.* Coll. Publ., London, 2020, pp. 87–112.

[3] F. Ferreira. 'Injecting uniformities into Peano arithmetic'. In: *Ann. Pure Appl. Logic* 157.2-3 (2009), pp. 122–129.

[4] F. Ferreira and P. Oliva. 'Bounded functional interpretation'. In: *Ann. Pure Appl. Logic* 135.1-3 (2005), pp. 73–112.

[5] 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.