Computational Interpretations of Analysis via Products of Selection Functions

Martín Escardó[1] and Paulo Oliva[2]

[1] University of Birmingham
[2] Queen Mary University of London

Abstract. We show that the computational interpretation of full comprehension via two well-known functional interpretations (dialectica and modified realizability) corresponds to two closely related infinite products of selection functions.

1 Introduction

Full classical analysis can be formalised using the language of finite types in Peano arithmetic PA^ω extended with the axiom schema of *full comprehension* (cf. [11])

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

As $\forall n^{\mathbb{N}}(A(n) \vee \neg A(n))$ is equivalent to $\forall n^{\mathbb{N}} \exists b^{\mathbb{B}}(b \leftrightarrow A(n))$, full comprehension, in the presence of classical logic, follows from *countable choice* over the booleans

$$\mathsf{AC}^{\mathbb{N}}_{\mathbb{B}} \quad : \quad \forall n^{\mathbb{N}} \exists b^{\mathbb{B}} A(n,b) \to \exists f \forall n A(n, fn).$$

Finally, the negative translation of $\mathsf{AC}^{\mathbb{N}}_{\mathbb{B}}$ follows intuitionistically from $\mathsf{AC}^{\mathbb{N}}_{\mathbb{B}}$ itself together with the classical principle of *double negation shift*

$$\mathsf{DNS} \quad : \quad \forall n^{\mathbb{N}} \neg\neg A(n) \to \neg\neg \forall n A(n),$$

where $A(n)$ can be assumed to be of the form[3] $\exists y B^N(n,y)$. Therefore, full classical analysis can be embedded (via the negative translation) into $\mathsf{HA}^\omega + \mathsf{AC}^{\mathbb{N}}_{\mathbb{B}} + \mathsf{DNS}$, where HA^ω is Heyting arithmetic in the language of all finite types. It then follows that a computational interpretation of theorems in analysis can be obtained via a computational interpretation of the theory $\mathsf{HA}^\omega + \mathsf{AC}^{\mathbb{N}}_{\mathbb{B}} + \mathsf{DNS}$. The fragment $\mathsf{HA}^\omega + \mathsf{AC}^{\mathbb{N}}_{\mathbb{B}}$, excluding the double negation shift, has a very straightforward (modified) realizability interpretation [15], as well as a dialectica interpretation [1, 10]. The remaining challenge is to give a computational interpretation of DNS.

A computational interpretation of DNS was first given by Spector [14], via the dialectica interpretation. Spector devised a form of recursion on well-founded trees, nowadays known as *bar recursion*, and showed that the dialectica interpretation of DNS can be witnesses by such recursion. A computational interpretation of DNS via realizability only came recently, first in [2], via a non-standard form of realizability, and then in [3, 4], via Kreisel's modified realizability. The realizability interpretation of DNS makes use of a new form of bar recursion, termed *modified bar recursion*.

[3] B^N being the (Gödel-Gentzen) negative translation of B.

In this article we show that both forms of bar recursion used to interpret classical analysis, via modified realizability and the dialectica interpretation, correspond to two closely related infinite products of selection functions [9].

Notation. We use X, Y, Z for variables ranging over types. Although in HA^ω one does not have dependent types, we will develop the rest of the paper working with types such as $\Pi_{i\in\mathbb{N}}X_i$ rather than its special case X^ω, when all X_i are the same. The reason for this generalisation is that all results below go through for the more general setting of dependent types. Nevertheless, we hesitate to define a formal extension of HA^ω with dependent types, leaving this to future work. We often write $\Pi_i X_i$ for $\Pi_{i\in\mathbb{N}}X_i$. Also, we write $\Pi_{i\geq k}X_i$ for $\Pi_i X_{k+i}$, and $\mathbf{0}$ for the constant functional 0 of a particular finite type. If α has type $\Pi_{i\in\mathbb{N}}X_i$ we use the following abbreviations

$$
\begin{aligned}
[\alpha](n) &\equiv \langle \alpha(0), \ldots, \alpha(n-1)\rangle, \quad \text{(initial segment of } \alpha \text{ of length } n) \\
\alpha[k,n] &\equiv \langle \alpha(k), \ldots, \alpha(n)\rangle, \quad \text{(finite segment from position } k \text{ to } n) \\
\overline{\alpha, n} &\equiv \langle \alpha(0), \ldots, \alpha(n-1), \mathbf{0}, \mathbf{0}, \ldots\rangle, \quad \text{(infinite extension of } [\alpha](n) \text{ with } \mathbf{0}\text{'s)} \\
\hat{s} &\equiv \langle s_0, \ldots, s_{|s|-1}, \mathbf{0}, \mathbf{0}, \ldots\rangle. \quad \text{(infinite extension of finite seq. } s \text{ with } \mathbf{0}\text{'s)}
\end{aligned}
$$

If x has type X_n and s has type $\Pi_{i=0}^{n-1}X_i$ then $s * x$ is the concatenation of s with x, which has type $\Pi_{i=0}^{n}X_i$. Similarly, if x has type X_0 and α has type $\Pi_{i=1}^{\infty}X_i$ then $x * \alpha$ has type $\Pi_{i\in\mathbb{N}}X_i$. Finally, by q_s or ε_s we mean the partial evaluation of q or ε on the finite string $s\colon \Pi_{i=0}^{n-1}X_i$, e.g. if q has type $\Pi_{i=0}^{\infty}X_i \to R$ then $q_s\colon \Pi_{i=n}^{\infty}X_i \to R$ is the functional $q_s(\alpha) = q(s * \alpha)$.

Acknowledgements. The second author gratefully acknowledges support of the Royal Society (grant 516002.K501/RH/kk).

1.1 Background: Selection functions and their binary product

In our recent paper [9] we showed how one can view any element of type $(X \to R) \to R$ as a *generalised quantifier*. The particular case when $R = \mathbb{B}$ corresponds to the types of the usual logical quantifiers $\forall, \exists$. We also showed that some generalised quantifiers $\phi\colon (X \to R) \to R$ are *attainable*, in the sense that for some *selection function* $\varepsilon\colon (X \to R) \to X$, we have

$$\phi p = p(\varepsilon p)$$

for all (generalised) predicates p. In the case when ϕ is the usual existential quantifier, for instance, ε corresponds to Hilbert's epsilon term. Since the types $(X \to R) \to R$ and $(X \to R) \to X$ shall be used quite often, we will abbreviate them as $K_R X$ and $J_R X$, respectively. Moreover, since R will be a fixed type, we often simply write KX and JX, omitting the subscript R. In [9] we also defined the following products of quantifiers and selection functions.

Definition 1. *Given a quantifier $\phi\colon KX$ and a family of quantifiers $\psi\colon X \to KY$, define a new quantifier $\phi \otimes \psi\colon K(X \times Y)$ as*

$$(\phi \otimes \psi)(p^{X\times Y\to R}) \stackrel{R}{:=} \phi(\lambda x^X.\psi(x, \lambda y^Y.p(x,y))).$$

Also, given a selection function $\varepsilon \colon JX$ *and a family of selection functions* $\delta \colon X \to JY$*, define a new selection function* $\varepsilon \otimes \delta \colon J(X \times Y)$ *as*

$$(\varepsilon \otimes \delta)(p^{X \times Y \to R}) \stackrel{X \times Y}{:=} (a, b(a))$$

where $b(x) := \delta(x, \lambda y^Y.p(x,y))$ *and* $a := \varepsilon(\lambda x^X.p(x, b(x)))$.

One of the results we obtained is that the product of attainable quantifiers is also attainable. This follows from the fact that the product of quantifiers corresponds to the product of selection functions, as made precise in the following lemma:

Lemma 1 ([9], lemma 3.1.2). *Given a selection function* $\varepsilon : JX$*, define a quantifier* $\overline{\varepsilon} \colon KX$ *as*

$$\overline{\varepsilon}p := p(\varepsilon p).$$

Then for $\varepsilon \colon JX$ *and* $\delta \colon X \to JY$ *we have* $\overline{\varepsilon \otimes \delta} = \overline{\varepsilon} \otimes \lambda x.\overline{\delta_x}$.

It is well known that the construction K can be given the structure of a strong monad, called the *continuation monad*. We have shown in [9] that J also is a strong monad, with the map $(\bar{\cdot}) \colon J \to K$ defined above playing the role of a monad morphism. Any strong monad T has a canonical morphism $TX \times TY \to T(X \times Y)$ (and a symmetric version). We have also shown in *loc. cit.* that for the monads $T = K$ and $T = J$ the canonical morphism turns out to be the product of quantifiers and of selection functions respectively. For further details on the connection between strong monads, products, and the particular monads J and K, see [9]. In the following we explore the concrete structure of J and K and their associated products considered as binary versions of bar recursion, which are then infinitely iterated to obtain countable versions.

2 Two Infinite Products of Selection Functions

Given a finite sequence of selection functions, the binary product defined above can be iterated so as to give rise to a finite product. We have shown that such construction appears in a variety of areas such as game theory (backward induction), algorithms (backtracking), and proof theory (interpretation of the infinite pigeon-hole principle). In the following we describe two possible ways of iterating the binary product of selection functions an infinite (or unbounded) number of times.

2.1 Explicitly controlled iteration

The first possibility for iterating the binary product of selection functions we consider here is via an "explicitly controlled" iteration, which we will show to correspond to Spector's bar recursion. In the following subsection we also define an "implicitly controlled" iteration, which we will show to correspond to modified bar recursion.

Definition 2. *Let* $\varepsilon \colon \Pi_{k\in\mathbb{N}}((\Pi_{j<k} X_j) \to JX_k)$ *be a family of selection functions. Define the* explicitly controlled infinite product of the selection functions ε *as*

$$\mathsf{EPS}_s(\omega)(\varepsilon) \overset{J(\Pi_{i=|s|}^{\infty} X_i)}{=} \begin{cases} \mathbf{0} & \text{if } \omega_s(\mathbf{0}) < |s| \\ \varepsilon_s \otimes \lambda x^{X_{|s|}}.\mathsf{EPS}_{s*x}(\omega)(\varepsilon) & \text{otherwise,} \end{cases}$$

where $s \colon \Sigma_{k\in\mathbb{N}}(\Pi_{j<k} X_j)$*. (Note that* $\omega_s(\mathbf{0}) = \omega(\hat{s})$*)*

We refer to this infinite iteration of the product $\otimes$ as "explicitly controlled" because we have an explicit test $\omega_s(\mathbf{0}) < |s|$ for when the iteration stops. As we will see in Section 2.2 (next), we could also iterate the product without using the functional ω.

As with Spector's bar recursion, we consider extensions of Gödel's T with the EPS-schema above. It is then natural to ask what are the models for the calculus of functionals $\mathsf{T} + \mathsf{EPS}$. It will follow from our result that EPS is primitive recursively equivalent to Spector's bar recursion, that EPS is validated both in the model of continuous functionals [13] and in the model of strongly majorizable functionals [6]. The same will be true for the functional IPS defined in Section 2.2. For further discussion on the models validating EPS and IPS see [9].

Lemma 2. *Let* $q \colon \Pi_{i=|s|}^{\infty} X_i \to R$ *and* $\omega \colon \Pi_i X_i \to \mathbb{N}$*.* EPS *can be equivalently defined as*

$$\mathsf{EPS}_s(\omega)(\varepsilon)(q) \overset{\Pi_{i=|s|}^{\infty} X_i}{=} \begin{cases} \mathbf{0} & \text{if } \omega_s(\mathbf{0}) < |s| \\ c * \mathsf{EPS}_{s*c}(\omega)(\varepsilon)(q_c) & \text{otherwise,} \end{cases}$$

where $c = \varepsilon_s(\lambda x.q_x(\mathsf{EPS}_{s*x}(\omega)(\varepsilon)(q_x)))$.

Although we will only need to work with EPS, it will be useful (for the sake of clarity) to define also the explicitly controlled infinite product of *quantifiers*:

Definition 3. *Let* $\phi \colon \Pi_{k\in\mathbb{N}}((\Pi_{j<k} X_j) \to KX_k)$ *be a family of quantifiers. The* explicitly controlled infinite product of the quantifiers ϕ *is defined as*

$$\mathsf{EPQ}_s(\omega)(\phi) \overset{K(\Pi_{i=|s|}^{\infty} X_i)}{=} \begin{cases} \lambda q.q(\mathbf{0}) & \text{if } \omega_s(\mathbf{0}) < |s| \\ \phi_s \otimes \lambda x^{X_{|s|}}.\mathsf{EPQ}_{s*x}(\omega)(\phi) & \text{otherwise.} \end{cases}$$

The following lemma explains why EPQ can be defined from EPS if we are working with attainable quantifiers.

Lemma 3. *Assuming* $\forall \alpha \exists n (\omega_{[\alpha](n)}(\mathbf{0}) \leq n)$ *we have* $\mathsf{EPQ}_s(\omega)(\overline{\varepsilon}) = \overline{\mathsf{EPS}_s(\omega)(\varepsilon)}$.

2.2 Implicitly controlled iteration

The binary product of selection functions can also be infinitely iterated without the need for the "control functional" ω as follows:

Definition 4. *Let* $\varepsilon \colon \Pi_{k\in\mathbb{N}}((\Pi_{j<k} X_j) \to (JX_k))$ *and* $s \colon \Sigma_{k\in\mathbb{N}}(\Pi_{j<k} X_j)$*. Define the* implicitly controlled infinite product of selection functions IPS *as*

$$\mathsf{IPS}_s(\varepsilon) \overset{J(\Pi_{i=|s|}^{\infty} X_i)}{=} \varepsilon_s \otimes \lambda x^{X_{|s|}}.\mathsf{IPS}_{s*x}(\varepsilon),$$

where $s \colon \Sigma_{k\in\mathbb{N}}(\Pi_{j<k} X_j)$.

Again, by unwinding the definition of the binary product of selection functions (and using course-of-values induction) one can show that IPS is equivalent to the following:

Lemma 4. *Let* $q\colon \Pi_{i=|s|}^{\infty} X_i \to R$. IPS *can be equivalently defined as*

$$\mathsf{IPS}_s(\varepsilon)(q)(n) \stackrel{X_{|s|+n}}{=} \varepsilon_{s*t_{s,n}}(\lambda x^{X_{|s|+n}}.q_{t_{s,n}*x}(\mathsf{IPS}_{s*t_{s,n}*x}(\varepsilon)(q_{t_{s,n}*x})))$$

where $t_{s,n} := [\mathsf{IPS}_s(\varepsilon)(q)](n)$.

The functional IPS generalises Escardó's [7] construction that selection functions for a sequence of spaces can be combined into a selection function for the product space.

Proposition 1. IPS *(with* $R = \mathbb{B}$ *and* ε_s *dependent only on* $|s|$*) is primitive recursively equivalent to Escardo's* Π *functional of [7]:*

$$\Pi(\varepsilon)(q)(n) \stackrel{X_n}{=} \varepsilon_n(\lambda x^{X_n}.q_{n,x}(\Pi(\lambda i.\varepsilon_{n+i+1})(q_{n,x})))$$

where

$$q_{n,x}(\alpha^{\Pi_{i=n+1}^{\infty} X_i}) \stackrel{\mathbb{B}}{:=} q\left(\lambda i. \begin{Bmatrix} \Pi(\varepsilon)(q)(i) & i < n \\ x & i = n \\ \alpha(i-n-1) & i > n \end{Bmatrix}\right).$$

Proof. For one direction we take

$$\Pi(\varepsilon)(q) := \mathsf{IPS}_{\langle\,\rangle}(\varepsilon)(q),$$

for the other

$$\mathsf{IPS}_s(\{\varepsilon_n\}_{n\in\mathbb{N}})(q) := \Pi(\{\varepsilon_{|s|+n}\}_{n\in\mathbb{N}})(q).$$

We omit the details of the verification. □

3 Dialectica Interpretation of Classical Analysis

We now show how EPS can be used to solve Spector's equations (which arise from the dialectica interpretation of full classical analysis).

Theorem 1 (cf. lemma 11.5 of [12]). *Let* $q\colon \Pi_{i=0}^{\infty} X_i \to R$ *and* $\omega\colon \Pi_{i=0}^{\infty} X_i \to \mathbb{N}$ *and* $\varepsilon\colon \Pi_{i=0}^{\infty} JX_i$ *be given. Define*

$$\begin{aligned} \alpha \quad &:= \mathsf{EPS}_{\langle\,\rangle}(\omega)(\varepsilon)(q) \\ p_n(x) &:= \mathsf{EPQ}_{[\alpha](n)*x}(\omega)(\overline{\varepsilon})(q_{[\alpha](n)*x}), \end{aligned}$$

identifying ε_s *with* $\varepsilon_{|s|}$. *The functionals* α *and* p_n *are a solution to Spector's system of equations, i.e. for* $n \leq \omega(\alpha)$ *we have*

$$\begin{aligned} \alpha(n) \quad &\stackrel{X_n}{=} \varepsilon_n(p_n) \\ p_n(\alpha(n)) &\stackrel{Y}{=} q\alpha. \end{aligned}$$

Proof. First, let us show by induction that for all n the following holds:

$$(i) \quad \alpha = [\alpha](n) * \mathsf{EPS}_{[\alpha](n)}(\omega)(\varepsilon)(q_{[\alpha](n)}).$$

If $n = 0$ this follows by definition. Assume this holds for n we wish to show it for $n+1$. Consider two cases.

(a) If $\omega(\overline{\alpha, n}) < n$ then $\mathsf{EPS}_{[\alpha](n)}(\omega)(\varepsilon)(q_{[\alpha](n)}) = \mathbf{0}$ and hence $\alpha \stackrel{\text{(IH)}}{=} \overline{\alpha, n} = \overline{\alpha, n+1}$. Therefore, $\omega(\overline{\alpha, n+1}) = \omega(\overline{\alpha, n}) < n < n+1$. So,

$$[\alpha](n+1) * \mathsf{EPS}_{[\alpha](n+1)}(\omega)(\varepsilon)(q_{[\alpha](n+1)}) = \overline{\alpha, n+1} = \overline{\alpha, n} = \alpha.$$

(b) If, on the other hand, $\omega(\overline{\alpha, n}) \geq n$, then

$$\alpha \stackrel{\text{(IH)}}{=} [\alpha](n) * \mathsf{EPS}_{[\alpha](n)}(\omega)(\varepsilon)(q_{[\alpha](n)}) = [\alpha](n) * c * \mathsf{EPS}_{[\alpha](n)*c}(\omega)(\varepsilon)(q_{[\alpha](n)*c}),$$

where $c = \alpha(n)$. Hence $\alpha = [\alpha](n+1) * \mathsf{EPS}_{[\alpha](n+1)}(\omega)(\varepsilon)(q_{[\alpha](n+1)})$.

Now, let $n := \omega(\alpha)$. We argue that (ii) $\omega(\overline{\alpha, n}) \geq n$. Otherwise, assuming $\omega(\overline{\alpha, n}) = \omega_{[\alpha](n)}(\mathbf{0}) < n$ we would have, by (i), that $\alpha = \overline{\alpha, n}$. Hence[4], $n > \omega_{[\alpha](n)}(\mathbf{0}) = \omega(\alpha) = n$, a contradiction.

Then, it follows easily that, if $n \leq \omega(\alpha)$,

$$\begin{aligned}
\alpha(n) &\stackrel{(i)}{=} \mathsf{EPS}_{[\alpha](n)}(\omega)(\varepsilon)(q_{[\alpha](n)})(0) \\
&\stackrel{(ii)}{=} (\varepsilon_n \otimes \lambda x.\mathsf{EPS}_{[\alpha](n)*x}(\omega)(\varepsilon))(q_{[\alpha](n)})(0) \\
&= \varepsilon_n(\lambda x.q_{[\alpha](n)*x}(\mathsf{EPS}_{[\alpha](n)*x}(\omega)(\varepsilon)(q_{[\alpha](n)*x}))) \\
&= \varepsilon_n(\lambda x.\overline{\mathsf{EPS}_{[\alpha](n)*x}(\omega)(\varepsilon)}(q_{[\alpha](n)*x})) \\
&= \varepsilon_n(\lambda x.\mathsf{EPQ}_{[\alpha](n)*x}(\omega)(\overline{\varepsilon})(q_{[\alpha](n)*x})) = \varepsilon_n(p_n).
\end{aligned}$$

For the second equality, we have

$$\begin{aligned}
p_n(\alpha(n)) &= \mathsf{EPQ}_{[\alpha](n+1)}(\omega)(\overline{\varepsilon})(q_{[\alpha](n+1)}) \\
&= \overline{\mathsf{EPS}_{[\alpha](n+1)}(\omega)(\varepsilon)}(q_{[\alpha](n+1)}) \\
&= q_{[\alpha](n+1)}(\mathsf{EPS}_{[\alpha](n+1)}(\omega)(\varepsilon)(q_{[\alpha](n+1)})) \\
&= q([\alpha](n+1) * \mathsf{EPS}_{[\alpha](n+1)}(\omega)(\varepsilon)(q_{[\alpha](n+1)})) \stackrel{(i)}{=} q(\alpha).
\end{aligned}$$

That concludes the proof. □

Remark 1. The theorem above has a very natural game theoretic reading. Following the nomenclature of [9], each ε_n can be viewed as the selection function defining an

[4] Note that a limited amount of extensionality is used here, which, nevertheless, can be avoided (cf. [12]). We recall that the dialectica interpretation does not validate the axiom of extensionality. We are obviously allowed, however, to appeal to extensionality when *verifying* that the dialectica interpretation of a certain principle (e.g. DNS) is correct.

outcome quantifier for round n. The functional q is the outcome functional, mapping infinite plays (in $\Pi_i X_i$) to the outcome of the game (in R). The construction used in the theorem for α and p_n calculates an infinite play α of the game which is optimal up to the point $n = \omega(\alpha)$. If ω is thought of as deciding when the game is terminated, then we have in fact an optimal play in the game.

Remark 2. Note that we are only using EPQ for the sake of clarity. As shown in Lemma 3, any use of EPQ above can be replaced by an instance of EPS. Therefore, the recursion schema EPS alone can be used to solve Spector's equations.

3.1 Relation to Spector's bar recursion

As we have shown above, EPS solves the computational interpretation of classical analysis via the dialectica interpretation. Spector, however, describing the recursion schema used in his solution, formulated first the general "construction by bar recursion" as

$$\mathsf{BR}_s(\omega)(\phi)(g) \stackrel{R}{=} \begin{cases} g(s) & \text{if } \omega_s(\mathbf{0}) < |s| \\ \phi_s(\lambda x^{X_{|s|}}.\mathsf{BR}_{s*x}(\omega)(\phi)(g)) & \text{otherwise.} \end{cases}$$

Then, Spector explicitly says that only a "restricted form" of this is used. It is this restricted form that we shall from now on call "Spector's bar recursion":

Definition 5. *Let* $R = \Pi_{i=0}^{\infty} X_i$ *and* $\varepsilon_s \colon JX_{|s|}$ *and* $\omega \colon \Pi_{i=0}^{\infty} X_i \to \mathbb{N}$. Spector's bar recursion *[14] is the following recursion schema*

$$\mathsf{SBR}_s(\omega)(\varepsilon) \stackrel{R}{=} \begin{cases} \hat{s} & \text{if } \omega_s(\mathbf{0}) < |s| \\ \mathsf{SBR}_{s*c}(\omega)(\varepsilon) & \text{otherwise,} \end{cases}$$

where $c \stackrel{X_{|s|}}{=} \varepsilon_s(\lambda x.\mathsf{SBR}_{s*x}(\omega)(\varepsilon))$.

We showed above how EPS can be used to solve Spector's equations. In fact, we have:

Proposition 2. EPS *and* SBR *are primitive recursively equivalent.*

4 Realizability Interpretation of Classical Analysis

We have seen (Section 3 above) that EPS solves the dialectica interpretation of classical analysis. In this section we show that when interpreting DNS via modified realizability, an *unrestricted* iterated product of selection functions naturally arises. Assuming continuity[5], for instance, one may say that the infinite iterated product is *implicitly* controlled, by the continuity of q.

As discussed in the introduction, only a restricted form of DNS is used for the interpretation of full comprehension, namely, DNS for formulas $A \equiv \exists y B^N(n, y)$. For such formulas we have that $\perp \to \forall n A(n)$, and hence this restricted form of DNS is equivalent to

[5] By continuity of $q \colon \Pi_i X_i \to R$ we mean that for all $\alpha \colon \Pi_i X_i$ there exists a point n (called 'point of continuity') such that the value $q(\alpha)$ is determined by $[\alpha](n)$, i.e. for any β extending $[\alpha](n)$ we have $q\alpha = q\beta$.

$$\forall n((A(n) \to \perp) \to A(n)) \to (\forall n A(n) \to \perp) \to \forall n A(n).$$

Moreover, since the negative translation brings us into minimal logic, falsity $\perp$ can be replaced by an arbitrary formula[6] R. In practice, however, because we will require a continuity assumption we restrict R to be a Σ^0_1 formula. As such, recalling that $J_R A \equiv (A \to R) \to A$, the resulting principle we obtain is what we shall call J-shift

$$J\text{-}\mathsf{shift} \quad : \quad \forall n J_R A(n) \to J_R \forall n A(n),$$

where $A(n)$ is an arbitrary formula and R is a Σ^0_1 formula.

Theorem 2 (cf. [3], theorem 3). $\mathsf{IPS}_{\langle\rangle}$ *modified realizes* J-shift.

Proof. Let

$$\varepsilon_n \ \mathsf{mr} \ (A(n) \to R) \to A(n)$$
$$q \ \ \mathsf{mr} \ \forall n A(n) \to R.$$

As in [3], we shall assume continuity of q. We show $\forall s \in S \, \forall n P(s, n)$ by relativised bar induction (see [3] for precise formulation), where

$$P(s, n) \equiv (s * \mathsf{IPS}_s(\varepsilon)(q_s))(n) \, \mathsf{mr} \, A(n)$$

and the predicate used in the relativisation is

$$s \in S \equiv \forall n < |s| \, (s_n \, \mathsf{mr} \, A(n)).$$

We write $\alpha \in S$ as an abbreviation for $\forall n([\alpha](n) \in S)$. We now prove the two assumptions of the bar induction:

(i) $\underline{\forall \alpha \in S \, \exists k \forall t \succeq [\alpha](k) \, \forall n P(t, n)}$, where $t \succeq s$ means that t is an extension of the finite sequence s. Given α we pick k to be a point of continuity of q on α. The result follows simply unfolding the definition of IPS.

(ii) $\underline{\forall s \in S(\forall t, x(s * t * x \in S \to \forall n P(s * t * x, n)) \to \forall n P(s, n))}$. Fix $s \in S$ and assume

(a) $\forall t, x(s * t * x \in S \to \forall n P(s * t * x, n))$.

We prove $\forall n P(s, n)$ by course-of-values induction. Assume $\forall k < n \, P(s, k)$, i.e.

(b) $\forall k < n \, ((s * \mathsf{IPS}_s(\varepsilon)(q_s))(k) \, \mathsf{mr} \, A(k))$.

We want to show $(s * \mathsf{IPS}_s(\varepsilon)(q_s))(n) \, \mathsf{mr} \, A(n)$. If $n < |s|$ we are done, since in this case $(s * \mathsf{IPS}_s(\varepsilon)(q_s))(n) = s_n$ (and $s \in S$). Assume $n \geq |s|$. Then, our goal becomes

$$\varepsilon_n(\lambda x^{X_n}.q_{s*t_{s,n}*x}(\mathsf{IPS}_{s*t_{s,n}*x}(\varepsilon)(q_{s*t_{s,n}*x}))) \, \mathsf{mr} \, A(n),$$

where $t_{s,n} = [\mathsf{IPS}_s(\varepsilon)(q_s)](n - |s|)$. That follows from

$$\lambda x^{X_n}.q_{s*t_{s,n}*x}(\mathsf{IPS}_{s*t_{s,n}*x}(\varepsilon)(q_{s*t_{s,n}*x})) \, \mathsf{mr} \, A(n) \to R$$

[6] This is known as the (refined) A-translation [5], and is useful to analyse proofs of Π^0_2 theorems in analysis.

which, by definition, is

$$\forall x^{X_n}(x \,\mathsf{mr}\, A(n) \;\to\; q_{s*t_{s,n}*x}(\mathsf{IPS}_{s*t_{s,n}*x}(\varepsilon)(q_{s*t_{s,n}*x})) \,\mathsf{mr}\, R).$$

Fix x such that $x \,\mathsf{mr}\, A(n)$. By our assumption (b) we have that $s * t_{s,n} * x \in S$. And by assumption (a) we get $(s * t_{s,n} * x * \mathsf{IPS}_{s*t_{s,n}*x}(\varepsilon)(q_{t_{s,n}*x})) \,\mathsf{mr}\, \forall n A(n)$. The proof is then concluded by the assumption that $q \,\mathsf{mr}\, \forall n A(n) \to R$. □

Remark 3. We analyse the J-shift in more detail in the companion paper [8], where a proof translation based on the construction JX is also defined.

4.1 Relation to modified bar recursion

We now show that IPS and modified bar recursion are in fact primitive recursively interdefinable. Modified bar recursion [3], when generalised to the language of dependent types, can be formulated as

$$\mathsf{MBR}_s(\varepsilon)(q)(n) \stackrel{X_n}{=} \begin{cases} s_n & \text{if } n < |s| \\ \varepsilon_s(\lambda x^{X_{|s|}}.q(\mathsf{MBR}_{s*x}(\varepsilon)(q)))(n-|s|) & \text{otherwise,} \end{cases}$$

where $\varepsilon_s \colon (X_{|s|} \to R) \to \Pi_{j \geq |s|} X_j$. The following lemma says that MBR is equivalent to a variant which can make use of any value bar recursively computed, and not just the immediate children $s * x$ of the node s. We are assuming that types are restricted so that finite sequences of X_k's can be coded as single elements.

Lemma 5 ([3], lemma 2). MBR *is primitive recursively equivalent to*

$$\mathsf{MBR}^0_s(\varepsilon)(q)(n) \stackrel{X_n}{=} \begin{cases} s_n & \text{if } n < |s| \\ \varepsilon_s(\lambda r^{\Pi_{k=|s|}^{j-1} X_k} \lambda x^{X_j}.q(\mathsf{MBR}^0_{s*r*x}(\varepsilon)(q)))(n-|s|) & \text{otherwise.} \end{cases}$$

The next theorem essentially says that MBR is also equivalent to a variant which makes use of course-of-values recursion to access values previously computed, i.e. in order to define the point n of the infinite sequence $\mathsf{MBR}^1_s(\varepsilon)(q)$ we are allowed to use $\mathsf{MBR}^1_s(\varepsilon)(q)(k)$ for $k < n$.

Lemma 6. MBR^0 *is primitive recursively equivalent to*

$$\mathsf{MBR}^1_s(\varepsilon)(q)(n) \stackrel{X_n}{=} \begin{cases} s_n & \text{if } n < |s| \\ \varepsilon_s(r_{s,n}, \lambda r^{\eta}, x^{X_j}.q(\mathsf{MBR}^1_{s*r*x}(\varepsilon)(q)))(n-|s|) & \text{otherwise,} \end{cases} \tag{1}$$

where $r_{s,n} := \mathsf{MBR}^1_s(\varepsilon)(q)[|s|, n-1]$ *and* $\eta = \Pi_{k=|s|}^{j-1} X_k$.

Corollary 1. MBR *primitive recursively defines* IPS.

Theorem 3. IPS *primitive recursively defines* MBR.

Corollary 2. *The equation defining* IPS *has a solution in the type structure* $\mathcal{M}$ *of the strongly majorizable functionals.*

Proof. This follows from the result in [4] that MBR lives in the model $\mathcal{M}$. □

We have also recently shown the following:

Theorem 4. *The iterated product of selection functions* $\bigotimes$ *defined in [9] (which is clearly a particular case of* IPS*) is primitive recursively equivalent to* IPS.

References

1. J. Avigad and S. Feferman. Gödel's functional ("Dialectica") interpretation. In S. R. Buss, editor, *Handbook of proof theory*, volume 137 of *Studies in Logic and the Foundations of Mathematics*, pages 337–405. North Holland, Amsterdam, 1998.
2. S. Berardi, M. Bezem, and T. Coquand. On the computational content of the axiom of choice. *The Journal of Symbolic Logic*, 63(2):600–622, 1998.
3. U. Berger and P. Oliva. Modified bar recursion and classical dependent choice. *Lecture Notes in Logic*, 20:89–107, 2005.
4. U. Berger and P. Oliva. Modified bar recursion. *Mathematical Structures in Computer Science*, 16:163–183, 2006.
5. U. Berger and H. Schwichtenberg. Program extraction from classical proofs. In D. Leivant, editor, *Logic and Computational Complexity Workshop (LCC'94)*, volume 960 of *Lecture Notes in Computer Science*, pages 77–97. Springer, Berlin, 1995.
6. M. Bezem. Strongly majorizable functionals of finite type: a model for bar recursion containing discontinuous functionals. *The Journal of Symbolic Logic*, 50:652–660, 1985.
7. M. H. Escardó. Infinite sets that admit fast exhaustive search. In *Proceedings of LICS*, pages 443–452, 2007.
8. M. H. Escardó and P. Oliva. The Peirce translation and the double negation shift. In F. Ferreira, B. Lowe, E. Mayordomo, and L. M. Gomes, editors, *Computability in Europe 2010, LNCS*. Springer, 2010.
9. M. H. Escardó and P. Oliva. Selection functions, bar recursion, and backward induction. *Mathematical Structures in Computer Science*, 20(2):127–168, 2010.
10. K. Gödel. Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes. *Dialectica*, 12:280–287, 1958.
11. U. Kohlenbach. Higher order reverse mathematics. In Stephen G. Simpson, editor, *Reverse Mathematics 2001*, volume 21 of *Lecture Notes in Logic*, pages 281–295. ASL, A K Peters, 2005.
12. U. Kohlenbach. *Applied Proof Theory: Proof Interpretations and their Use in Mathematics*. Monographs in Mathematics. Springer, 2008.
13. D. Normann. The continuous functionals. In E. R. Griffor, editor, *Handbook of Computability Theory*, chapter 8, pages 251–275. North Holland, Amsterdam, 1999.
14. C. Spector. Provably recursive functionals of analysis: a consistency proof of analysis by an extension of principles in current intuitionistic mathematics. In F. D. E. Dekker, editor, *Recursive Function Theory: Proc. Symposia in Pure Mathematics*, volume 5, pages 1–27. American Mathematical Society, Providence, Rhode Island, 1962.
15. A. S. Troelstra. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*, volume 344 of *Lecture Notes in Mathematics*. Springer, Berlin, 1973.