BAR RECURSION AND PRODUCTS OF SELECTION FUNCTIONS

MARTÍN ESCARDÓ AND PAULO OLIVA

Abstract. We show how two iterated products of selection functions can both be used in conjunction with system T to interpret, via the dialectica interpretation and modified realizability, full classical analysis. We also show that one iterated product is equivalent over system T to Spector's bar recursion, whereas the other is T-equivalent to modified bar recursion. Modified bar recursion itself is shown to arise directly from the iteration of a different binary product of 'skewed' selection functions. Iterations of the dependent binary products are also considered but in all cases are shown to be T-equivalent to the iteration of the simple products.

§**1. Introduction.** Gödel's [13] so-called dialectica interpretation reduces the consistency of Peano arithmetic to the consistency of the quantifier-free calculus of functionals T. In order to extend Gödel's interpretation to full classical analysis $\mathrm{PA}^\omega + \mathrm{CA}$, Spector [19] made use of the fact that $\mathrm{PA}^\omega + \mathrm{CA}$ can be embedded, via the negative translation, into $\mathrm{HA}^\omega + \mathrm{AC}_{\mathbb{N}} + \mathrm{DNS}$. Here PA^ω and HA^ω denote Peano and Heyting arithmetic, respectively, formulated in the language of finite types, and

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

is *full comprehension*,

$$\mathrm{AC}_{\mathbb{N}} : \ \forall n^{\mathbb{N}} \exists x^{X} A(n,x) \to \exists f \forall n A(n, fn)$$

is *countable choice*, and

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

is the *double negation shift*, with $A(n)$ and $A(n,x)$ standing for arbitrary formulas, and $B(n) \equiv \exists x \neg A(n,x)$. Since $\mathrm{HA}^\omega + \mathrm{AC}_{\mathbb{N}}$, excluding the double negation shift, has a straightforward (modified) realizability interpretation [21], as well as a dialectica interpretation [1, 13], the remaining challenge is to give a computational interpretation to DNS.

A computational interpretation of DNS was first given by Spector [19], 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 witnessed by such kind of recursion. A computational interpretation of DNS via realizability only came recently, first in [2], via a non-standard form of realizability, and then in [4, 5], via Kreisel's modified realizability. The realizability interpretation of DNS makes use of a new form of bar recursion, termed *modified bar recursion*.

Received by the editors PREPRINT, APRIL 16, 2013.

It has been shown in [5] that Spector's bar recursion is definable in system T extended with modified bar recursion, but not conversely, since Spector's bar recursion is S1-S9 computable in the model of total continuous functionals, but modified bar recursion is not.

In the present paper we revisit these functional interpretations of classical analysis from the perspective of the newly developed theory of selection functions [8, 9, 10, 11]. *Selection functionals* are functionals of type $(X \to R) \to X$, for arbitrary finite types X, R. We think of mappings $p\colon X \to R$ as generalised predicates, and of functionals $\varepsilon\colon (X \to R) \to X$ as witnessing, when possible, the "non-emptiness" of any given such predicate. For instance, if $R = \mathbb{B}$ is the set of booleans, Hilbert's ε-constant can be viewed as a selection function. Just as ε-terms in Hilbert's calculus can be used to define the existential quantifier, so can any selection function $\varepsilon\colon (X \to R) \to X$ be used to define a *generalised quantifier* $\phi\colon (X \to R) \to R$ as

$$\phi(p) \stackrel{R}{=} p(\varepsilon(p)).$$

Moreover, just like the usual quantifiers $\exists^X$ and $\forall^Y$ can be nested to produce a quantifier on the product space $X \times Y$, so can generalised quantifiers and selection functions. We prefer to think about the nesting of selection functions (and quantifiers) as a *product operation*, since it transform selection functions over spaces X and Y into a new selection function on the product space $X \times Y$ (cf. [11]).

In this article we define two different iterations of the binary product of selection functions, one which we call *implicitly controlled* and the other which we call *explicitly controlled*. We show that:

- Modified bar recursion is T-equivalent to the implicitly controlled product of selection functions.
- Spector's bar recursion is T-equivalent to the explicitly controlled product of selection functions.
- The two different products can be used to interpret DNS directly via modified realizability and the dialectica interpretation, respectively.
- The implicitly controlled product of selection functions is strictly stronger than the explicitly controlled one.
- Apparently stronger iterations of the dependent products are in fact T-equivalent to the iterations of the simple products.

§**2. Preliminaries.** Before we present our main results, let us first define the formal systems used, and give an introduction to our recent work on selection functions.

2.1. Heyting arithmetic and system T. In this section we define the formal system used to prove the inter-definability results. These include Heyting arithmetic in all finite types and extensions including bar induction and a continuity principle.

DEFINITION 2.1 (Finite types). *The set of all* finite types $\mathcal{T}$ *are defined inductively as*

- $\mathbb{B}$ *(booleans) and* $\mathbb{N}$ *(integers) are in* $\mathcal{T}$

- *If X and Y are in $\mathcal{T}$ then $X \times Y$ (product) and $X \to Y$ (functions) are in $\mathcal{T}$*
- *If X is in $\mathcal{T}$ then X^* (finite sequence) is in $\mathcal{T}$.*

We will also make informal use of the following type construction: Given a sequence of types $(X_i)_{i\in\mathbb{N}}$ we also consider $\Pi_{i\in\mathbb{N}} X_i$ as a type. The main purpose of this is to make the constructions more readable, since we can keep track of the positions which are being changed. A formal extension of system T with such type construction has been considered by Tait [20], hence we also hope that our presentation below will extend smoothly to a more general setting, although in this paper we focus on the standard version of system T.

We use X, Y, Z for variables ranging over the elements of $\mathcal{T}$. We often write $\Pi_i X_i$ for $\Pi_{i\in\mathbb{N}} X_i$, and also $\Pi_{i\geqslant k} X_i$ for $\Pi_i X_{i+k}$.

Let HA^ω be usual Heyting arithmetic in all finite types with a fully extensional treatment of equality, as in the system E-HA^ω of [21]. Its quantifier-free fragment is the usual Gödel's system T, also extended with sequence types. Gödel's primitive recursion for each sequence of types $(X_i)_{i\in\mathbb{N}} \in \mathcal{T}$ is given by

$$\begin{array}{lcl} \mathsf{R}fg0 & \overset{X_0}{=} & g \\ \mathsf{R}fg(n+1) & \overset{X_{n+1}}{=} & fn(\mathsf{R}fgn) \end{array}$$

where R has finite type $\Pi_n(X_n \to X_{n+1}) \to X_0 \to \Pi_i X_i$. If the reader prefers, however, she can assume that all X_i are equal X and read $\Pi_{i\in\mathbb{N}} X_i$ as X^ω. We also assume that we have a constant $\mathbf{0}^X$ of each finite type X, and the usual constructors and destructors such as $\langle t^X, s^Y\rangle\colon X \times Y$ and $\pi_i(\langle s_0^{X_0}, s_1^{X_1}\rangle) = s_i$, where $i = \{0, 1\}$, for instance. For the newly introduced sequence types we have that if $t\colon \Pi_i X_i$ then $ti\colon X_i$; and if $t\colon X_i$ then $\lambda i.t\colon \Pi_i X_i$. If $s\colon \Pi_{i<n} X_i$, we write $s_i\colon X_i$ for the i-th element of the sequence, for $i < n$. If $s\colon \Pi_{i<n}(X_i \times Y_i)$ is a sequence of pairs, we write $s^0\colon \Pi_{i<n} X_i$ and $s^1\colon \Pi_{i<n} Y_i$ for the projection of the sequence on the first and second coordinates, respectively. If α has type $\Pi_{i\in\mathbb{N}} X_i$ we use the following abbreviations

$$\begin{array}{lcl} \alpha^n & \equiv & \lambda i.\alpha(n+i), \quad \text{(the } n\text{-left shift of } \alpha\text{, hence } \alpha^n\colon \Pi_i X_{n+i}) \\ q^n(\alpha) & \equiv & q(\alpha^n), \quad (\text{so } q^n\colon \Pi_i X_i \to R \text{ if } q\colon \Pi_i X_{n+i} \to R) \\ \alpha[k, n] & \equiv & \langle \alpha(k), \dots, \alpha(n)\rangle, \quad \text{(finite segment from position } k \text{ to } n) \\ [\alpha](n) & \equiv & \alpha[0, n-1], \quad \text{(initial segment of } \alpha \text{ of length } n) \\ \overline{\alpha, n} & \equiv & \langle \alpha(0), \dots, \alpha(n-1), \mathbf{0}, \mathbf{0}, \dots\rangle, \quad \text{(infinite extension of } [\alpha](n) \text{ with } \mathbf{0}\text{'s)} \end{array}$$

where in the last case the type of $\mathbf{0}$ at the i-th coordinate is the same type of $\alpha(i)$. If x has type X_n and s has type $\Pi_{i<n} X_i$ then $s * x$ is the concatenation of s with x, which has type $\Pi_{i<n+1} X_i$. Similarly, if x has type X_0 and α has type $\Pi_i X_{i+1}$ then $x * \alpha$ has type $\Pi_i X_i$.

In the following we shall assume that certain types are *discrete*. Semantically, in the model of total continuous functionals, discreteness means that singletons are open or that all points are isolated. Syntactically, the following grammar produces discrete types in that model (along with compact types) [8].

DEFINITION 2.2 (Discrete and compact types). *Define the two subsets of $\mathcal{T}$ inductively as follows:*

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

For the first part of the paper, up to the end of Section 5, we work with a model independent notion of definability. Formally, given a term t in system T, we view an equation $F(x) = t(F, x)$ as *defining* or *specifying* a functional F. We do not worry whether such an equation has a solution in any particular model of HA^ω, or whether it is unique, when it has a solution. After this general model-independent development we consider particular models in the final section, and prove some non-definability results. The two main models we will consider are that of partial/total continuous functionals [17], and strongly majorizable functionals [7].

DEFINITION 2.3. *We say that a functional G is T-definable from a functional F (written $G \leqslant_T F$) over a theory $\mathcal{S}$ if there exists a term s in system T such that $s(F)$ satisfies the defining equation of G provably in $\mathcal{S}$. We say that F and G are T-equivalent over $\mathcal{S}$, written $F =_T G$, if $G \geqslant_T F$ and $F \geqslant_T G$.*

When stating in a theorem or proposition that G is T-definable in F, we will explicitly write after the theorem/proposition number the theory $\mathcal{S}$ that is need for the verification. In a few cases this theory will be an extension of HA^ω with the following two principles: *Spector's condition*

$$\mathrm{SPEC}: \ \forall \omega^{\Pi_i X_i \to \mathbb{N}} \forall \alpha^{\Pi_i X_i} \exists n (\omega(\overline{\alpha, n}) < n),$$

and the scheme of *relativised bar induction* BI

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

where $S(s)$ and $P(s)$ are arbitrary predicates in the language of HA^ω, and $\alpha \in S$ and $s \in S$ are shorthands for $\forall n S([\alpha](n))$ and $S(s)$ respectively. When $P(s)$ is restricted to be quantifier-free we write $\mathrm{BI}_{\mathrm{QF}}$.

We note that SPEC follows from the *axiom of continuity*

$$\mathrm{CONT}: \ \forall q^{\Pi_i X_i \to R} \forall \alpha \exists n \forall \beta ([\alpha](n) \stackrel{\Pi_{i \leq n} X_i}{=} [\beta](n) \to q(\alpha) \stackrel{R}{=} q(\beta))$$

with R discrete, but it also holds in the model of strongly majorizable functionals [7].

2.2. Selection functions and generalised quantifiers. In [11] we have studied the properties of functionals having the type $(X \to R) \to R$, and called these *generalised quantifiers*. When $R = \mathbb{B}$ we have that $(X \to \mathbb{B}) \to \mathbb{B}$ is the type 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 $(X \to R) \to X$ will be used quite often, we abbreviate them as $K_R X$ and $J_R X$, respectively. Moreover, when R is fixed, we often simply write KX and JX, omitting the subscript R. In [11] we also defined products of quantifiers and selection functions.

DEFINITION 2.4 (Product of selection functions and quantifiers). *Given generalised quantifiers $\phi\colon KX$ and $\psi\colon KY$, define the product quantifier $(\phi \otimes \psi)\colon K(X \times Y)$ as*

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

Also, given selection functions $\varepsilon\colon JX$ and $\delta\colon JY$, define the product 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

$$\begin{array}{lcl} a & \stackrel{X}{=} & \varepsilon(\lambda x^X.p(x,b(x))) \\ b(x^X) & \stackrel{Y}{=} & \delta(\lambda y^Y.p(x,y)). \end{array}$$

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 2.5 ([11], lemma 3.1.2). *Let R be fixed. 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 $\delta\colon JY$ we have $\overline{\varepsilon \otimes \delta} = \overline{\varepsilon} \otimes \overline{\delta}$.

Given a finite sequence of selection functions or quantifiers, the two binary products defined above can be iterated so as to give rise to finite products of selection functions and quantifiers. We have shown that such a construction also appears in game theory (backward induction), algorithms (backtracking), and proof theory (interpretation of the infinite pigeon-hole principle) – see [11] for details.

In the following (Sections 3 and 4) we will describe two possible ways of iterating the binary product of selection function an infinite, or unbounded, number of times.

§3. Explicitly Controlled Product.

The finite product of selection functions of Definition 2.4 can be infinitely iterated in two ways. The first, which we define in this section is via an *explicitly controlled* iteration, which we will show to correspond to Spector's bar recursion. In the following section we also define an *implicitly controlled* iteration, which we will show to correspond to modified bar recursion.

DEFINITION 3.1 (eps). *Let* $\varepsilon\colon \Pi_k JX_k$ *be a sequence of selection functions. Define their* explicitly controlled infinite product *as*

$$\mathsf{eps}^l_n(\varepsilon)(q) \overset{\Pi_i X_{i+n}}{=} \begin{cases} \mathbf{0} & \text{if } l(q(\mathbf{0})) < n \\ (\varepsilon_n \otimes \mathsf{eps}^l_{n+1}(\varepsilon))(q) & \text{otherwise,} \end{cases} \tag{1}$$

where $q\colon \Pi_i X_{i+n} \to R$ *and* $l\colon R \to \mathbb{N}$. *We call* l *the* length function *since it controls the length of the recursive path.*

Given a sequence of selection functions $\varepsilon\colon \Pi_k JX_k$, at stage n we apply the binary product of selection functions to ε_n and the result of the recursive call at stage $n+1$. That is done until the condition $l(q(\mathbf{0})) < n$ is met. In order to see why such a condition is eventually met (assuming continuity, for instance) it is best to look at the following equivalent formulation of eps.

LEMMA 3.2 (HA^ω). *Let* $q\colon \Pi_i X_{i+n} \to R$ *and* $l\colon R \to \mathbb{N}$. *The functional* eps *can be equivalently defined as*

$$\mathsf{eps}^l_n(\varepsilon)(q) \overset{\Pi_i X_{i+n}}{=} \begin{cases} \mathbf{0} & \text{if } l(q(\mathbf{0})) < n \\ c * \mathsf{eps}^l_{n+1}(\varepsilon)(q_c) & \text{otherwise,} \end{cases}$$

where $c = \varepsilon_n(\lambda x.\overline{\mathsf{eps}^l_{n+1}(\varepsilon)}(q_x))$; *or even more generally*

$$\mathsf{eps}^l_n(\varepsilon)(q)(i) \overset{X_{i+n}}{=} \begin{cases} \mathbf{0} & \text{if } \exists s \preceq t(l(q_s(\mathbf{0})) < n + |s|) \\ \varepsilon_{i+n}(\lambda x.\overline{\mathsf{eps}^l_{i+n+1}(\varepsilon)}(q_{t*x})) & \text{otherwise,} \end{cases}$$

where $t = [\mathsf{eps}^l_n(\varepsilon)(q)](i)$.

PROOF. The first equivalent formulation is obtained by simply unfolding of binary product of selection functions (Definition 2.4). For the second formulation one uses course-of-values induction on i noticing that as soon as the condition $l(q_s(\mathbf{0})) < n + |s|$ is satisfied for some s then the value of $\mathsf{eps}^l_n(\varepsilon)(q)(i)$ will be $\mathbf{0}$ for $i \geqslant |s|$. ⊣

The fact that eps exists in the model of total continuous functionals, and is in fact uniquely characterized by its defining equation, can be seen as follows. First, note that the $\mathsf{eps}_n(\varepsilon)(q)$ is an infinite sequence, say $\alpha\colon \Pi_i X_{i+n}$. Intuitively, at each recursive call the functional q gets information about one more element of its input sequence. Assuming continuity we will have that $l \circ q\colon \Pi_{i+n} X_i \to \mathbb{N}$ will eventually always return a fixed value, no matter what the rest of the input sequence is. This means that as n increases we will eventually have $l(q(\mathbf{0})) < n$. It is perhaps surprising that such a functional also exists in the model of strongly majorizable functionals [7], which contains discontinuous functionals! Following the construction of Bezem [7] one can prove this directly, but this result will also follow from our result that eps is T-definable from Spector's bar recursion (Section 3.3).

We also define the corresponding explicitly controlled product of quantifiers as follows:

DEFINITION 3.3 (epq). *Let* $\phi\colon \Pi_k K X_k$ *be a sequence of quantifiers. Their* explicitly controlled infinite product *is defined as*

$$\mathsf{epq}^l_n(\phi)(q) \stackrel{R}{=} \begin{cases} q(\mathbf{0}) & \text{if } l(q(\mathbf{0})) < n \\ (\phi_n \otimes \mathsf{epq}^l_{n+1}(\phi))(q) & \text{otherwise,} \end{cases}$$

where $q\colon \Pi_i X_{i+n} \to R$ *and* $l\colon R \to \mathbb{N}$. *Unfolding the definition of the binary product of quantifiers we have*

$$\mathsf{epq}^l_n(\phi)(q) \stackrel{R}{=} \begin{cases} q(\mathbf{0}) & \text{if } l(q(\mathbf{0})) < n \\ \phi_n(\lambda x^{X_n}.\mathsf{epq}^l_{n+1}(\phi)(q_x)) & \text{otherwise.} \end{cases} \tag{2}$$

Note that under the assumption of epq we can derive Spector's condition SPEC. The same proofs as given in lemma 3C of [14], observing that the instance of bar recursion needed there can be easily defined from epq.

LEMMA 3.4. $\mathrm{HA}^\omega + (2) \vdash \mathrm{SPEC}$.

First, we show that in terms of T-definability epq is stronger than eps. Although care has to be taken when spelling out the details, the proof essentially makes use of the fact that each selection function ε defines a quantifier, as $\phi(p) = p(\varepsilon(p))$ (cf. Lemma 2.5).

THEOREM 3.5 ($\mathrm{HA}^\omega + \mathrm{BI}$). $\mathsf{epq} \geqslant_T \mathsf{eps}$.

PROOF. In order to define eps for the types (X_i, R) we shall use epq for the types $(X_i, \Pi_i X_i)$. The explicitly controlled product of quantifiers is related to Spector's general form of bar recursion as the explicitly controlled product of selection functions is related to Spector's "restricted form" of bar recursion (cf. [18, 19]). Hence, the proof presented here that epq defines eps is essentially the same as Spector's proof that his restricted form of bar recursion follows from the general form. Let us abbreviate $R' = \Pi_i X_i$. Given $\varepsilon_i\colon J_R X_i$ and $q\colon \Pi_i X_i \to R$ define $\phi_i^{\varepsilon,q}\colon K_{R'} X_i$ as

$$\phi_i^{\varepsilon,q}(p^{X_i \to R'}) \stackrel{R'}{=} p(\varepsilon_i(\lambda x^{X_i}.q(p(x)))). \tag{3}$$

Given $\alpha\colon \Pi_i X_i$ let $\alpha^n\colon \Pi_{i \geqslant n} X_i$ be

$$\alpha^n(i) = \alpha(i+n).$$

The construction α^n drops the first n elements of α. Given $q\colon \Pi_{i \geqslant n} X_i \to R$ let $q^n\colon \Pi_i X_i \to R$ be

$$q^n(\alpha^{\Pi_i X_i}) \stackrel{R}{=} q(\alpha^n).$$

Hence, q^n behaves as q except that it ignores the first n elements of its input sequence. We claim that eps can be defined from epq as

$$\mathsf{eps}^l_n(\varepsilon)(q) \stackrel{\Pi_i X_{i+n}}{=} (\mathsf{epq}^{l \circ q^n}_n(\phi^{\varepsilon,q^n})(\lambda \alpha^{\Pi_i X_{n+i}}.\mathbf{0}^{\Pi_{i<n} X_i} * \alpha))^n. \tag{4}$$

We consider two cases. If $l(q(\mathbf{0}^{\Pi_{i \geqslant n} X_i})) < n$ then we also have $l(q^n(\mathbf{0}^{\Pi_i X_i})) < n$. Therefore

$$\begin{aligned} \mathsf{eps}^l_n(\varepsilon)(q)(i) &\stackrel{(4)}{=} \mathsf{epq}^{l \circ q^n}_n(\phi^{\varepsilon,q^n})(\lambda \alpha.\mathbf{0}^{\Pi_{i<n} X_i} * \alpha)(n+i) \\ &\stackrel{(2)}{=} \mathbf{0}^{X_{n+i}}. \end{aligned}$$

So, $\mathsf{eps}^l_n(\varepsilon)(q) = \mathbf{0}^{\Pi_i X_{n+1}}$ as desired.
On the other hand, if $l(q(\mathbf{0}^{\Pi_{i\geqslant n}X_i})) \geqslant n$, since this implies $l(q^n(\mathbf{0}^{\Pi_i X_i})) \geqslant n$, and we have:

$$\begin{array}{rcl}
\mathsf{eps}^l_n(\varepsilon)(q) & \overset{(4)}{=} & (\mathsf{epq}^{l\circ q^n}_n(\phi^{\varepsilon,q^n})(\lambda\alpha.\mathbf{0}^{\Pi_{i<n}X_i} * \alpha))^n \\
& \overset{(2)}{=} & (\phi^{\varepsilon,q^n}_n(\lambda x^{X_n}.\mathsf{epq}^{l\circ q^n}_{n+1}(\phi^{\varepsilon,q^n})((\lambda\alpha.\mathbf{0}^{\Pi_{i<n}X_i} * \alpha)_x)))^n \\
& = & (\phi^{\varepsilon,q^n}_n(\lambda x^{X_n}.\mathsf{epq}^{l\circ q^n}_{n+1}(\phi^{\varepsilon,q^n})(\lambda\alpha.\mathbf{0}^{\Pi_{i<n}X_i} * x * \alpha)))^n \\
& \overset{(3)}{=} & (\mathsf{epq}^{l\circ q^n}_{n+1}(\phi^{\varepsilon,q^n})(\lambda\alpha.\mathbf{0}^{\Pi_{i<n}X_i} * c * \alpha))^n \\
& \overset{(+)}{=} & (((\mathbf{0} * c) @ \mathsf{epq}^{l\circ(q_c)^{n+1}}_{n+1}(\phi^{\varepsilon,(q_c)^{n+1}})(\lambda\alpha.\mathbf{0}^{\Pi_{i<n+1}X_i} * \alpha)))^n \\
& = & (c * (\mathsf{epq}^{l\circ(q_c)^{n+1}}_{n+1}(\phi^{\varepsilon,(q_c)^{n+1}})(\lambda\alpha.\mathbf{0}^{\Pi_{i<n+1}X_i} * \alpha))^{n+1}) \\
& \overset{(4)}{=} & (c * \mathsf{eps}^l_{n+1}(\varepsilon)(q_c))
\end{array}$$

where

$$\begin{array}{rcl}
c & = & \varepsilon_n(\lambda x^{X_n}.q^n(\mathsf{epq}^{l\circ q^n}_{n+1}(\phi^{\varepsilon,q^n})(\lambda\alpha.\mathbf{0}^{\Pi_{i<n}X_i} * x * \alpha))) \\
& \overset{(+)}{=} & \varepsilon_n(\lambda x^{X_n}.q^n((\mathbf{0} * x) @ \mathsf{epq}^{l\circ(q_x)^{n+1}}_{n+1}(\phi^{\varepsilon,(q_x)^{n+1}})(\lambda\alpha.\mathbf{0}^{\Pi_{i<n+1}X_i} * \alpha))) \\
& = & \varepsilon_n(\lambda x^{X_n}.(q_x)^{n+1}(\mathsf{epq}^{l\circ(q_x)^{n+1}}_{n+1}(\phi^{\varepsilon,(q_x)^{n+1}})(\lambda\alpha.\mathbf{0}^{\Pi_{i<n+1}X_i} * \alpha))) \\
& = & \varepsilon_n(\lambda x^{X_n}.q_x(\mathsf{epq}^{l\circ(q_x)^{n+1}}_{n+1}(\phi^{\varepsilon,(q_x)^{n+1}})(\lambda\alpha.\mathbf{0}^{\Pi_{i<n+1}X_i} * \alpha))^{n+1}) \\
& \overset{(4)}{=} & \varepsilon_n(\lambda x^{X_n}.q_x(\mathsf{eps}^l_{n+1}(\varepsilon)(q_x))).
\end{array}$$

Property

$$(+)\ \mathsf{epq}^{l\circ q^n}_{n+1}(\phi^{\varepsilon,q^n})(\lambda\alpha.\mathbf{0} * x * \alpha) = (\mathbf{0} * x) @ \mathsf{epq}^{l\circ(q_x)^{n+1}}_{n+1}(\phi^{\varepsilon,(q_x)^{n+1}})(\lambda\alpha.\mathbf{0} * \alpha)$$

follows by BI + SPEC. We take $S(s) = \mathsf{true}$ and

$$\underbrace{\mathsf{epq}^{l\circ q^n}_{n+j+1}(\phi^{\varepsilon,q^n})(\lambda\alpha.\mathbf{0} * x * s * \alpha) = (\mathbf{0} * x) @ \mathsf{epq}^{l\circ(q_x)^{n+1}}_{n+j+1}(\phi^{\varepsilon,(q_x)^{n+1}})(\lambda\alpha.\mathbf{0} * s * \alpha)}_{P(s)}$$

where $s\colon \Pi_{n<i\leqslant n+j}X_i$, so $|s| = j$, and $q\colon \Pi_{i\geqslant n}X_i \to R$ so $q^n\colon \Pi_i X_i \to R$.
By SPEC, for any $\alpha\colon \Pi_{i>n}X_i$ there is a point j such that

$$\begin{array}{rcl}
(l\circ q^n)(\mathbf{0} * x * [\alpha](j) * \beta) & = & l(q(x * [\alpha](j) * \beta)) \\
& < & n + j + 1.
\end{array}$$

For such j and $s = [\alpha](j)$ it is easy to see that $P(s)$ holds as both sides of $P(s)$ are equal to $\mathbf{0} * x * s * \mathbf{0}$. That proves the base case of BI.

Assume now that (IH) $\forall y P(s*y)$ holds and let us prove $P(s)$. We have

$$\begin{aligned}
&\mathsf{epq}^{l\circ q^n}_{n+j+1}(\phi^{\varepsilon,q^n})(\lambda\alpha.\mathbf{0}*x*s*\alpha)\\
&\quad\overset{(2)}{=}\phi^{\varepsilon,q^n}_{n+j+1}(\lambda y.\mathsf{epq}^{l\circ q^n}_{n+j+2}(\phi^{\varepsilon,q^n})(\lambda\alpha.\mathbf{0}*x*s*y*\alpha))\\
&\quad\overset{(\mathrm{IH})}{=}\phi^{\varepsilon,q^n}_{n+j+1}(\lambda y.((\mathbf{0}*x)\mathbin{@}\mathsf{epq}^{l\circ(q_x)^{n+1}}_{n+j+2}(\phi^{\varepsilon,(q_x)^{n+1}})(\lambda\alpha.\mathbf{0}*s*y*\alpha)))\\
&\quad\overset{(3)}{=}(\mathbf{0}*x)\mathbin{@}\mathsf{epq}^{l\circ(q_x)^{n+1}}_{n+j+2}(\phi^{\varepsilon,(q_x)^{n+1}})(\lambda\alpha.\mathbf{0}*s*c*\alpha)\\
&\quad=(\mathbf{0}*x)\mathbin{@}\mathsf{epq}^{l\circ(q_x)^{n+1}}_{n+j+2}(\phi^{\varepsilon,(q_x)^{n+1}})(\lambda\alpha.\mathbf{0}*s*\tilde{c}*\alpha)\\
&\quad=(\mathbf{0}*x)\mathbin{@}\phi^{\varepsilon,(q_x)^{n+1}}_{n+j+1}(\lambda y.\mathsf{epq}^{l\circ(q_x)^{n+1}}_{n+j+2}(\phi^{\varepsilon,(q_x)^{n+1}})(\lambda\alpha.\mathbf{0}*s*y*\alpha))\\
&\quad\overset{(3)}{=}(\mathbf{0}*x)\mathbin{@}\mathsf{epq}^{l\circ(q_x)^{n+1}}_{n+j+1}(\phi^{\varepsilon,(q_x)^{n+1}})(\lambda\alpha.\mathbf{0}*s*\alpha)
\end{aligned}$$

where

$$\begin{aligned}
c &:= \varepsilon_{n+j+1}(\lambda y.q^n((\mathbf{0}*x)\mathbin{@}\mathsf{eps}^{l\circ(q_x)^{n+1}}_{n+j+2}(\phi^{\varepsilon,(q_x)^{n+1}})(\lambda\alpha.\mathbf{0}*s*y*\alpha)))\\
&= \varepsilon_{n+j+1}(\lambda y.(q_x)^{n+1}(\mathsf{eps}^{l\circ(q_x)^{n+1}}_{n+j+2}(\phi^{\varepsilon,(q_x)^{n+1}})(\lambda\alpha.\mathbf{0}*s*y*\alpha)))\\
&=: \tilde{c}.
\end{aligned}$$

That concludes the proof. ⊣

Hence, we have shown that eps is T-definable in epq. The converse, that epq is T-definable in eps has been recently shown in [18].

3.1. Dialectica interpretation of classical analysis. In order to find witnesses for the dialectica interpretation of DNS, and hence full classical analysis, Spector arrived at the following system of equations

$$\begin{aligned}
n &\overset{\mathbb{N}}{=} \omega\alpha,\\
\alpha(n) &\overset{X}{=} \varepsilon_n(p),\\
p(\alpha(n)) &\overset{R}{=} q\alpha,
\end{aligned}\tag{5}$$

where $\varepsilon_n\colon J_R X$ and $q\colon(\mathbb{N}\to X)\to R$ and $\omega\colon(\mathbb{N}\to X)\to\mathbb{N}$ are given and $n\colon\mathbb{N}$ and $\alpha\colon\mathbb{N}\to X$ and $p\colon X\to R$ are the unknowns. We now show how eps can be used to solve Spector's equations. We first solve a slightly different set of equations, and as a corollary we obtain a solution to Spector's original one.

THEOREM 3.6 ($\mathrm{HA}^\omega+(1)$). *Let $q\colon\Pi_i X_i\to R$ and $l\colon R\to\mathbb{N}$ and $\varepsilon\colon\Pi_i J_R X_i$ be given. Define*

$$\begin{aligned}
\alpha &= \mathsf{eps}^l_0(\varepsilon)(q)\\
p_n(x) &= \overline{\mathsf{eps}^l_{n+1}(\varepsilon)}(q_{[\alpha](n)*x}).
\end{aligned}$$

For $n\leqslant l(q(\alpha))$ we have

$$\begin{aligned}
\alpha(n) &\overset{X_n}{=} \varepsilon_n(p_n)\\
p_n(\alpha(n)) &\overset{R}{=} q\alpha.
\end{aligned}$$

PROOF. This is essentially Spector's proof (cf. lemma 11.5 of [16]). First, let us show by induction that for all n the following holds:

$(i)\quad \alpha = [\alpha](n) * \mathsf{eps}^l_n(\varepsilon)(q_{[\alpha](n)})$.

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

(a) If $l(q_{[\alpha](n)}(\mathbf{0})) = l(q(\overline{\alpha,n})) < n$ then $\mathsf{eps}^l_n(\varepsilon)(q_{[\alpha](n)}) = \mathbf{0}$ and hence

$\alpha \overset{\text{(IH)}}{=} \overline{\alpha,n} = \overline{\alpha,n+1}$.

Therefore, $l(q(\overline{\alpha,n+1})) = l(q(\overline{\alpha,n})) < n < n+1$. So,

$[\alpha](n+1) * \mathsf{eps}^l_{n+1}(\varepsilon)(q_{[\alpha](n+1)}) = \overline{\alpha,n+1} = \overline{\alpha,n} = \alpha$.

(b) If, on the other hand, $l(q_{[\alpha](n)}(\mathbf{0})) = l(q(\overline{\alpha,n})) \geqslant n$, then

$$\alpha \overset{\text{(IH)}}{=} [\alpha](n) * \mathsf{eps}^l_n(\varepsilon)(q_{[\alpha](n)}) = [\alpha](n) * c * \mathsf{eps}^l_{n+1}(\varepsilon)(q_{[\alpha](n)*c}),$$

where $c = \alpha(n)$. Hence $\alpha = [\alpha](n+1) * \mathsf{eps}^l_{n+1}(\varepsilon)(q_{[\alpha](n+1)})$. That proves (i).

Now, assume $n \leqslant l(q(\alpha))$. We first argue that (ii) $n \leqslant l(q(\overline{\alpha,n}))$. Otherwise, assuming $n > l(q(\overline{\alpha,n})) = l(q_{[\alpha](n)}(\mathbf{0}))$ we would have, by (i), that $\alpha = \overline{\alpha,n}$. And hence, by extensionality, $n > l(q_{[\alpha](n)}(\mathbf{0})) = l(q(\alpha)) \geqslant n$, which is a contradiction.

Hence, assuming $n \leqslant l(q(\alpha))$ we have

$$\begin{aligned}
\alpha(n) &\overset{(i)}{=} \mathsf{eps}^l_n(\varepsilon)(q_{[\alpha](n)})(0)\\
&\overset{(ii)}{=} (\varepsilon_n \otimes \mathsf{eps}^l_{n+1}(\varepsilon))(q_{[\alpha](n)})(0)\\
&= \varepsilon_n(\lambda x.q_{[\alpha](n)*x}(\mathsf{eps}^l_{n+1}(\varepsilon)(q_{[\alpha](n)*x})))\\
&= \varepsilon_n(\lambda x.\overline{\mathsf{eps}^l_{n+1}(\varepsilon)}(q_{[\alpha](n)*x}))\\
&= \varepsilon_n(p_n).
\end{aligned}$$

For the second equality, we have

$$\begin{aligned}
p_n(\alpha(n)) &= \overline{\mathsf{eps}^l_{n+1}(\varepsilon)}(q_{[\alpha](n+1)})\\
&= q_{[\alpha](n+1)}(\mathsf{eps}^l_{n+1}(\varepsilon)(q_{[\alpha](n+1)}))\\
&= q([\alpha](n+1) * \mathsf{eps}^l_{n+1}(\varepsilon)(q_{[\alpha](n+1)})) \overset{(i)}{=} q(\alpha).
\end{aligned}$$

⊣

COROLLARY 3.7. *For any given*

$$\begin{aligned}
\varepsilon_n &: J_R X\\
q &: X^{\mathbb{N}} \to R\\
\omega &: X^{\mathbb{N}} \to \mathbb{N},
\end{aligned}$$

there are $\alpha\colon \mathbb{N} \to X$ *and* $p\colon X \to R$ *satisfying equation (5).*

PROOF. Let $R' = R \times \mathbb{N}$, and let $\pi_0 \colon R \times \mathbb{N} \to R$ and $\pi_1 \colon R \times \mathbb{N} \to \mathbb{N}$ denote the first and second projections. Define

$$\begin{aligned} q'(\alpha) &\overset{R'}{=} \langle q(\alpha), \omega(\alpha) \rangle \\ \varepsilon'_n(p^{X \to R'}) &\overset{X}{=} \varepsilon_n(\lambda x^X.\pi_0(p(x))). \end{aligned}$$

so $q' \colon (\mathbb{N} \to X) \to R'$ and $\varepsilon'_n \colon J_{R'}X$. Let

$$\begin{aligned} \alpha &\overset{X^{\mathbb{N}}}{=} \mathsf{eps}_0^{\pi_1}(\varepsilon')(q') \\ p'_n(x^X) &\overset{R'}{=} \overline{\mathsf{eps}_{n+1}^{\pi_1}(\varepsilon')}(q'_{[\alpha](n)*x}). \end{aligned}$$

Assume $n \leqslant \omega(\alpha) = \pi_1(q'(\alpha))$. By Theorem 3.6 we have

$$\begin{aligned} \alpha(n) &\overset{X}{=} \varepsilon'_n(p'_n) \\ p'_n(\alpha(n)) &\overset{R'}{=} q'\alpha. \end{aligned}$$

Finally, let $n = \omega(\alpha)$ and $p(x) = p_n(x) = \pi_0(p'_n(x))$. Then α and p satisfy the desired equation, since $\varepsilon'_n(p'_n) = \varepsilon_n(p_n)$. ⊣

3.2. Dependent variants of eps **and** epq**.** We have considered in previous papers [9, 11] a slight generalisation of the product of selection functions, where a selection function (or a quantifier) at stage n can have access to the previously computed values X_i for $i < n$. We called this the *dependent* product of selection functions and quantifiers.

DEFINITION 3.8 (Dependent product of selection functions and quantifiers). *Given a quantifier $\phi \colon KX$ and a family of quantifiers $\psi \colon X \to KY$, define the dependent product quantifier $(\phi \otimes_d \psi) \colon K(X \times Y)$ as*

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

Also, given a selection function $\varepsilon \colon JX$ and a family of selection functions $\delta \colon X \to JY$, define the dependent product selection function $(\varepsilon \otimes_d \delta) \colon J(X \times Y)$ as

$$(\varepsilon \otimes_d \delta)(p^{X \times Y \to R}) \overset{X \times Y}{=} (a, b(a))$$

where

$$\begin{aligned} a &\overset{X}{=} \varepsilon(\lambda x^X.p(x, b(x))) \\ b(x) &\overset{Y}{=} \delta(x, \lambda y^Y.p(x,y)). \end{aligned}$$

As done for the simple product of selection functions and quantifiers, we can also iterate the dependent products as follows:

DEFINITION 3.9 (EPS and EPQ). *Given a family of selection functions*

$$\varepsilon \colon \Pi_k(\Pi_{i<k}X_i \to JX_k),$$

define their dependent explicitly controlled product *(denoted* EPS*) as*

$$\mathsf{EPS}_s^l(\varepsilon)(q) \overset{\Pi_i X_{|s|+i}}{=} \begin{cases} \mathbf{0} & \text{if } l(q(\mathbf{0})) < |s| \\ (\varepsilon_s \otimes_d (\lambda x^{X_{|s|}}.\mathsf{EPS}_{s*x}^l(\varepsilon)))(q) & \text{otherwise.} \end{cases} \tag{6}$$

As done for eps *in Lemma 3.2,* EPS *can also be equivalently formulated as*

$$\mathsf{EPS}^l_s(\varepsilon)(q)(i) \overset{X_{|s|+i}}{=} \begin{cases} \mathbf{0} & \text{if } \exists r \preceq t(l(q_r(\mathbf{0})) < |s| + |r|) \\ \varepsilon_{s*t}(\lambda x^{X_{|s*t|}}.\overline{\mathsf{EPS}^l_{s*t*x}(\varepsilon)}(q_{t*x})) & \text{otherwise,} \end{cases}$$

where $t = [\mathsf{EPS}^l_s(\varepsilon)(q)](i)$.

Moreover, given a family of quantifiers

$$\phi\colon \Pi_k(\Pi_{i<k}X_i \to KX_k),$$

define their dependent explicitly controlled product *(denoted* EPQ*) as*

$$\mathsf{EPQ}^l_s(\phi)(q) \overset{\Pi_i X_{|s|+i}}{=} \begin{cases} q(\mathbf{0}) & \text{if } l(q(\mathbf{0})) < |s| \\ (\phi_s \otimes_d (\lambda x^{X_{|s|}}.\mathsf{EPQ}^l_{s*x}(\phi)))(q) & \text{otherwise.} \end{cases} \tag{7}$$

In the dialectica interpretation of DNS given above (Section 3.1), the selection functions ε_n do not depend on the history of choices already made. Thus, it was sufficient to use an iteration of the *simple* product of selection functions. Nevertheless, Spector bar recursion and modified bar recursion are normally formulated in the most general form, where selection functions at point n have access to the values $i < n$.

Clearly eps is T-definable from EPS. We now show that in fact eps and EPS are T-equivalent.

THEOREM 3.10 (HA^ω). $\mathsf{eps} \geqslant_T \mathsf{EPS}$.

PROOF. To define EPS of type (X_k, R) we use eps of type $(\Pi_{i<k}X_i \to X_k, R)$. The idea is to make use of a construction of type

$$(X \to JY) \to J(X \to Y)$$

that turns a family of selection function into a single selection function of a (possibly) higher type level. Hence, a family of selection functions $\varepsilon_k\colon \Pi_{i<k}X_i \to JX_k$ can be turned into a sequence of selection functions $\tilde{\varepsilon}_k\colon J(\Pi_{i<k}X_i \to X_k)$ as

$$\tilde{\varepsilon}_k(P^{(\Pi_{i<k}X_i \to X_k)\to R}) \overset{\Pi_{i<k}X_i \to X_k}{=} \lambda s^{\Pi_{i<k}X_i}.\varepsilon_s(\lambda y^{X_k}.P(\lambda t.y)).$$

Given $l\colon R \to \mathbb{N}$, the infinite (simple) product of selection functions gives

$$\mathsf{eps}^l_n(\tilde{\varepsilon}) \quad : \quad J(\Pi_{i\geqslant n}(\Pi_{k<i}X_k \to X_i)).$$

The dependent product can then be defined as

$$\mathsf{EPS}^l_s(\varepsilon)(q^{\Pi_i X_{|s|+i}\to R}) \overset{\Pi_i X_{|s|+i}}{=} (\mathsf{eps}^l_{|s|}(\tilde{\varepsilon})(q^s))^s,$$

where $s\colon \Pi_{k<j}X_k$ and $\alpha^s(i) \overset{X_{|s|+i}}{=} \alpha(i)(s * [\alpha^s](i))$ and $q^s(\alpha) = q(\alpha^s)$. We must show that EPS as defined above satisfies the defining equation (6). We do this by course-of-values induction on i. Let

$$t = [\mathsf{EPS}^l_s(\varepsilon)(q)](i) \overset{(\mathrm{IH})}{=} [(\mathsf{eps}^l_{|s|}(\tilde{\varepsilon})(q^s))^s](i),$$

where $q\colon \Pi_i X_{|s|+i} \to R$ and $q^s\colon \Pi_k(\Pi_{i<|s|+k}X_i \to X_{|s|+k}) \to R$. If for some $u \preceq t$ we have $l(q_u(\mathbf{0})) < |s| + |u|$ the result is trivial, because $(q^s)_{u'}(\mathbf{0}) = q_u(\mathbf{0})$ for some $u' \preceq t' = [\mathsf{eps}^l_{|s|}(\tilde{\varepsilon})(q^s)](i)$, and hence $(\mathsf{eps}^l_{|s|}(\tilde{\varepsilon})(q^s))^s(i) = \mathbf{0}$. On the other

hand, assuming $\forall u \preceq t\ l(q_u(\mathbf{0})) \geqslant |s| + |u|$, unfolding definitions we have: Abbreviating $u = [\mathsf{eps}^l_{|s|}(\tilde{\varepsilon})(q^s)](i)$

$$\begin{aligned}
\mathsf{EPS}^l_s(\varepsilon)(q)(i) &\overset{X_{|s|+i}}{=} (\mathsf{eps}^l_{|s|}(\tilde{\varepsilon})(q^s))^s(i)\\
&= \mathsf{eps}^l_{|s|}(\tilde{\varepsilon})(q^s)(i)(s*t)\\
&\overset{\text{L3.2}}{=} \tilde{\varepsilon}_{|s|+i}(\lambda f.(q^s)_{u*f}(\mathsf{eps}^l_{|s|+i+1}(\tilde{\varepsilon})((q^s)_{u*f})))(s*t)\\
&= \varepsilon_{s*t}(\lambda x.(q^s)_{u*\lambda r.x}(\mathsf{eps}^l_{|s|+i+1}(\tilde{\varepsilon})((q^s)_{u*\lambda r.x})))\\
&\overset{(i,ii)}{=} \varepsilon_{s*t}(\lambda x.q_{t*x}(\mathsf{eps}^l_{|s|+i+1}(\tilde{\varepsilon})((q^s)_{u*\lambda r.x}))^{s*t*x})\\
&\overset{(i)}{=} \varepsilon_{s*t}(\lambda x.q_{t*x}(\mathsf{eps}^l_{|s|+i+1}(\tilde{\varepsilon})((q_{t*x})^{s*t*x}))^{s*t*x})\\
&= \varepsilon_{s*t}(\lambda x.q_{t*x}(\mathsf{EPS}^l_{s*t*x}(\varepsilon)(q_{t*x})))\\
&= \varepsilon_{s*t}(\lambda x.\overline{\mathsf{EPS}^l_{s*t*x}(\varepsilon)}(q_{t*x})).
\end{aligned}$$

We have used the following facts which are easy to verify

(i) $(t*\alpha)^s = t^s * (\alpha)^{s*t^s}$, for all t, s, α,

(ii) $u^s = t$, for t, s and u as above.

⊣

QUESTION 3.11. *Note that a similar construction does not work in the case of quantifiers, since there is no λ-term of type $(X \to KY) \to K(X \to Y)$, for arbitrary X and Y. In fact, in the case of quantifiers it is still open whether the simple explicitly controlled iteration is T-equivalent to the explicitly controlled iteration of the dependent product of quantifiers.*

3.3. Relation to Spector's bar recursion. As we have shown in Theorem 3.6, which is essentially Spector's solution, the explicitly controlled product of selection functions eps can also be used to give a computational interpretation of classical analysis. When presenting his solution in [19], Spector first formulates a general "construction by bar recursion" as

$$\mathsf{BR}^\omega_s(\phi)(q) \overset{R}{=} \begin{cases} q_s(\mathbf{0}) & \text{if } \omega_s(\mathbf{0}) < |s| \\ \phi_s(\lambda x^{X_{|s|}}.\mathsf{BR}^\omega_{s*x}(\phi)(q)) & \text{otherwise,} \end{cases}$$

where $\phi_s \colon K_R X_{|s|}$, $q \colon \Pi_i X_i \to R$ and $\omega \colon \Pi_i X_i \to \mathbb{N}$. This is usually referred to as *Spector's bar recursion*, but we argue that this is misleading. We show that BR is closely related to the product of *quantifiers* EPQ, whereas the special case of this used by Spector is equivalent to the (dependent) product of *selection functions* EPS, which we have shown to be equivalent to eps (Section 3.2).

REMARK 3.12. *In fact, Spector's definition seems slightly more general than* BR *as defined here, since in Spector's definition q might also depend on the length of the sequence s. As we show in Lemma 5.1, however, it is possible to reconstruct $|s|$ from the sequence $s * \mathbf{0}$ if s is the point where Spector's condition first happens.*

THEOREM 3.13 ($\mathsf{HA}^\omega + \mathsf{BI}$). $\mathsf{BR} \geqslant_T \mathsf{EPQ}$.

PROOF. In order to define EPQ of type (X_i, R) we use BR of the same type (X_i, R). BR and EPQ has very similar definitions, except that in BR the stopping condition is given directly on the current sequence $s * \mathbf{0}$, whereas in EPQ a "length" function $l\colon R \to \mathbb{N}$ is used so that the stopping condition involves the composition $l \circ q$. Hence, in order to define EPQ from BR it is essentially enough to take $\omega = l \circ q$, taking care of the fact that the types of q in EPQ and BR are slightly different as q in EPQ takes a "shorter" input sequence starting at point $|s|$. Define

$$\mathsf{EPQ}^l_s(\phi)(q) = \mathsf{BR}^{l\circ q^{|s|}}_s(\phi)(q^{|s|}).$$

If $(l \circ q^{|s|})_s(\mathbf{0}) = l(q(\mathbf{0})) < |s|$ then

$$\mathsf{EPQ}^l_s(\phi)(q) = \mathsf{BR}^{l\circ q^{|s|}}_s(\phi)(q^{|s|}) = (q^{|s|})_s(\mathbf{0}) = q(\mathbf{0}).$$

On the other hand, if $(l \circ q^{|s|})_s(\mathbf{0}) = l(q_s(\mathbf{0})) \geqslant |s|$ then

$$\begin{aligned}
\mathsf{EPQ}^l_s(\phi)(q) &= \mathsf{BR}^{l\circ q^{|s|}}_s(\phi)(q^{|s|}) \\
&= \phi_s(\lambda x^{X_{|s|}}.\mathsf{BR}^{l\circ q^{|s|}}_{s*x}(\phi)(q^{|s|})) \\
&\overset{(+)}{=} \phi_s(\lambda x^{X_{|s|}}.\mathsf{BR}^{l\circ (q_x)^{|s*x|}}_{s*x}(\phi)((q_x)^{|s*x|}) \\
&= \phi_s(\lambda x^{X_{|s|}}.\mathsf{EPQ}^l_{s*x}(\phi)(q_x)) \\
&= (\phi_s \otimes \lambda x^{X_{|s|}}.\mathsf{EPQ}^l_{s*x}(\phi))(q)
\end{aligned}$$

where $(+)$ $\mathsf{BR}^{l\circ q^{|s|}}_{s*x}(\phi)(q^{|s|}) = \mathsf{BR}^{l\circ (q_x)^{|s*x|}}_{s*x}(\phi)((q_x)^{|s*x|})$ can, as in Theorem 3.5, be proved by SPEC and BI, since $q^{|s|}(s * x * \alpha) = (q_x)^{|s*x|}(s * x * \alpha)$. Recall that $\mathrm{HA}^\omega + \mathsf{BR} \vdash \mathsf{SPEC}$ (cf. Lemma 3.4). ⊣

Spector, however, explicitly says that only a *restricted form* of BR is used for the dialectica interpretation of (the negative translation of) countable choice. It is this restricted form that we shall from now on call *Spector's bar recursion*:

DEFINITION 3.14 (Spector's bar recursion). *Spector's bar recursion [19] is the recursion schema*

$$\mathsf{SBR}^\omega_s(\varepsilon) \overset{\Pi_i X_i}{=} s @ \begin{cases} \mathbf{0} & \text{if } \omega_s(\mathbf{0}) < |s| \\ \mathsf{SBR}^\omega_{s*c}(\varepsilon) & \text{otherwise,} \end{cases} \tag{8}$$

where $c \overset{X_{|s|}}{=} \varepsilon_s(\lambda x^{X_{|s|}}.\mathsf{SBR}^\omega_{s*x}(\varepsilon))$, *and where* $\varepsilon_s\colon J_{\Pi_i X_i} X_{|s|}$ *and* $\omega\colon \Pi_i X_i \to \mathbb{N}$.

We now show that Spector's bar recursion is T-definable from the explicitly controlled product of selection functions EPS. It will follow for other results that they are in fact T-equivalent (see Figure 1).

THEOREM 3.15 (HA^ω). $\mathsf{EPS} \geqslant_T \mathsf{SBR}$.

PROOF. To define SBR of type (X_i) we use EPS of type $(X_i, (\Pi_i X_i) \times \mathbb{N})$. EPS and SBR have very similar definitions, except that EPS has an extra argument $q\colon \Pi_{i>|s|} X_i \to R$. We can obtain SBR from EPS by simply taking $q(\alpha)$ to be the identity function plus the stopping value $\omega(\alpha)$. So, the length function $l\colon R \to$

$\mathbb{N}$ can be taken to the the second projection. The details are as follows: Let $R = \Pi_i X_i \times \mathbb{N}$. Given $\omega \colon \Pi_i X_i \to \mathbb{N}$ and $\varepsilon_s \colon J_{\Pi_i X_i} X_{|s|}$, define

$$\begin{aligned} l(r^R) &\overset{\mathbb{N}}{=} \pi_1(r) \\ q^\omega(\alpha^{\Pi_i X_i}) &\overset{R}{=} \langle \alpha, \omega(\alpha) \rangle \\ \tilde{\varepsilon}_s(p^{X_{|s|} \to R}) &\overset{X_{|s|}}{=} \varepsilon_s(\pi_0 \circ p). \end{aligned}$$

Define

$$\mathsf{SBR}_s^\omega(\varepsilon) \overset{\Pi_i X_i}{=} s * \mathsf{EPS}_s^l(\tilde{\varepsilon})(q_s^\omega).$$

If $\omega_s(\mathbf{0}) < |s|$ then

$$\begin{aligned} \mathsf{SBR}_s^\omega(\varepsilon) &= s * \mathsf{EPS}_s^l(\tilde{\varepsilon})(q_s^\omega) \\ &= s * \mathbf{0} \\ &= s \mathbin{@} \mathbf{0}. \end{aligned}$$

Assume now that $\omega_s(\mathbf{0}) \geqslant |s|$. We then have

$$\begin{aligned} \mathsf{SBR}_s^\omega(\varepsilon) &= s * \mathsf{EPS}_s^l(\tilde{\varepsilon})(q_s^\omega) \\ &= s * c * \mathsf{EPS}_{s*c}^l(\tilde{\varepsilon})(q_{s*c}^\omega) \\ &= \mathsf{SBR}_{s*c}^\omega(\varepsilon) \end{aligned}$$

where

$$\begin{aligned} c &= \tilde{\varepsilon}_s(\lambda x.q_{s*x}^\omega(\mathsf{EPS}_{s*x}^l(\varepsilon)(q_{s*x}^\omega))) \\ &= \varepsilon_s(\lambda x.s * x * \mathsf{EPS}_{s*x}^l(\varepsilon)(q_{s*x}^\omega)) \\ &= \varepsilon_s(\lambda x.\mathsf{SBR}_{s*x}^\omega(\varepsilon)) \end{aligned}$$

which concludes the proof. ⊣

§**4. Implicitly Controlled Product.** We have seen in Section 3 above that the explicitly controlled iterated product of selection functions is sufficient to witness the dialectica interpretation of the double negation shift (and hence, classical countable choice). In this section we show that when interpreting this same principle via modified realizability, one seems to need an *unrestricted* or, as we we shall call it, *implicitly controlled* infinite product of selection functions.

DEFINITION 4.1 (ips). *The* implicitly controlled product *of a family* $\varepsilon \colon \Pi_k J X_k$ *of selection functions is defined as*

$$\mathsf{ips}_n(\varepsilon) \overset{J(\Pi_i X_{i+n})}{=} \varepsilon_n \otimes \mathsf{ips}_{n+1}(\varepsilon).$$

Unfolding the definition of $\otimes$*, this is the same as*

$$\mathsf{ips}_n(\varepsilon)(q) \overset{\Pi_i X_{i+n}}{=} \underbrace{\varepsilon_n(\lambda x.q_x(\mathsf{ips}_{n+1}(\varepsilon)(q_x)))}_{c} * \mathsf{ips}_{n+1}(\varepsilon)(q_c). \tag{9}$$

We call the above infinite product *implicitly controlled* because under the assumption of continuity for functionals of type $\Pi_i X_{i+n} \to R$, for discrete R, the bar recursive calls eventually terminate. Unwinding the definition of the binary product, ips can also be equivalently formulated as follows.

LEMMA 4.2 (HA^ω). *The functional* ips *can be equivalently defined by the equation*

$$\mathsf{ips}_n(\varepsilon)(q)(i) \overset{X_{n+i}}{=} \varepsilon_{n+i}(\lambda x^{X_{n+i}}.\overline{\mathsf{ips}_{n+i+1}(\varepsilon)}(q_{s*x}))$$

where $q\colon \Pi_i X_{n+i} \to R$ *and* $s = [\mathsf{ips}_n(\varepsilon)(q)](i)$.

PROOF. By unfolding the definition of the binary product of selection functions using course-of-values induction. ⊣

4.1. Realizability interpretation of classical analysis. We now describe how ips can be used to interpret the double negation shift (and hence countable choice) via modified realizability. As discussed in the introduction, a computational interpretation of full classical analysis can be reduced to an interpretation of the double negation shift DNS. Given that the formula $A(n)$ (in DNS) can be assumed to be of the form $\exists x \neg B(n,x)$, DNS is equivalent to

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

That is because, for $A(n) \equiv \exists x \neg B(n,x)$, we have both $\bot \to A(n)$ and $\bot \to \forall n A(n)$ in *minimal logic*. Moreover, since the negative translation brings us into minimal logic, falsity $\bot$ can be replaced by an arbitrary Σ^0_1-formula R. This is known as the (refined) A-translation [6], and is useful to analyse proofs of Π^0_2 theorems in analysis. Recall that we are using the abbreviation

$$J_R A = (A \to R) \to A.$$

The resulting principle we obtain is what we shall call the J-shift

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

DNS is then the particular case of the K-shift

$$K\text{-shift} \quad : \quad \forall n K_R A(n) \to K_R \forall n A(n),$$

when $R = \bot$; considering the other type construction

$$K_R A = (A \to R) \to R.$$

One advantage of moving to the J-shift is that $A(n)$ now can be taken to be an arbitrary formula, not necessarily of the form $\exists x \neg B(n,x)$. Hence the principle J-shift is more general than DNS. We analyse the logical strength of the principle J-shift in more detail in [10], where a proof translation based on the construction $J_R A$ is also defined. Our proof of the following theorem is very similar to that of [4, Theorem 3]. We assume continuity and relativised bar induction as formulated in Section 2.1.

THEOREM 4.3 ($\mathrm{HA}^\omega + \mathrm{BI} + \mathrm{CONT}$). ips_0 *modified realizes* J-shift.

PROOF. Given a term t and a formula A we write "$t\,\mathsf{mr}\,A$" for "t modified realizes A" (see [21] for definition). Assume that

$$\varepsilon_n \quad \mathsf{mr} \quad (A(n) \to R) \to A(n),$$

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

We show $\forall s \in S\, \forall n P(s,n)$ by relativised bar induction, 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

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

Formally, we are making use of an equivalent so-called *monotone* variant of BI in which the conclusion $P(\langle\,\rangle)$ can be strengthened to $\forall s \in S\, P(s)$, given that we can prove $\forall \alpha^S \exists k \forall t \succeq [\alpha](k) P(t)$ instead of $\forall \alpha^S \exists k P([\alpha](k))$.
We write $\alpha \in S$ as an abbreviation for $\forall n([\alpha](n) \in S)$. The first assumption of BI (i.e. $S(\langle\,\rangle)$) is vacuously true. We now prove the other two assumptions.

(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 the point of continuity of q on α. The result follows simply by unfolding the definition of ips.

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

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

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

$$\text{(b) } \forall k < n\, ((s * \mathsf{ips}_{|s|}(\varepsilon)(q_s))(k) \,\mathsf{mr}\, A(k)).$$

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 $n \geqslant |s|$. Then our goal becomes

$$\varepsilon_n(\lambda x^{X_n}.q_{s*t*x}(\mathsf{ips}_{n+1}(\varepsilon)(q_{s*t*x}))) \,\mathsf{mr}\, A(n),$$

where $t = [\mathsf{ips}_{|s|}(\varepsilon)(q_s)](n - |s|)$. This follows from

$$\lambda x^{X_n}.q_{s*t*x}(\mathsf{ips}_{n+1}(\varepsilon)(q_{s*t*x})) \,\mathsf{mr}\, A(n) \to R$$

which, by definition, is

$$\forall x^{X_n}(x \,\mathsf{mr}\, A(n) \;\to\; q_{s*t*x}(\mathsf{ips}_{n+1}(\varepsilon)(q_{s*t*x})) \,\mathsf{mr}\, R).$$

Let x such that $x \,\mathsf{mr}\, A(n)$. By our assumption (b) we have that $s * t * x \in S$. And by assumption (a) we get $(s * t * x * \mathsf{ips}_{n+1}(\varepsilon)(q_{s*t*x})) \,\mathsf{mr}\, \forall n A(n)$. The proof is then concluded by the assumption that $q \,\mathsf{mr}\, \forall n A(n) \to R$. ⊣

4.2. Dependent variant of ips**.** The proof given for the equivalence between EPS and eps in Section 3.2 can be easily adapted to show that also ips is T-equivalent to its dependent variant IPS.

DEFINITION 4.4 (IPS). *Let* $\varepsilon\colon \Pi_k(\Pi_{i<k}X_i \to JX_k)$. *Define the* dependent implicitly controlled product of selection functions *(denoted* IPS*) as*

$$\mathsf{IPS}_s(\varepsilon) \overset{J(\Pi_i X_{|s|+i})}{=} \varepsilon_s \otimes_d (\lambda x^{X_{|s|}}.\mathsf{IPS}_{s*x}(\varepsilon)).$$

As in Lemma 4.2 for ips*, we can also equivalently define* IPS *as*

$$\mathsf{IPS}_s(\varepsilon)(q)(i) \overset{X_{|s|+i}}{=} \varepsilon_{s*t}(\lambda x^{X_{|s|+i}}.\overline{\mathsf{IPS}_{s*t*x}(\varepsilon)}(q_{t*x})),$$

where $t = [\mathsf{IPS}_s(\varepsilon)(q)](i)$.

We again use the fact that a family of selection functions $\Pi_{i<k}X_i \to JX_k$ can be turned into a single section function $J(\Pi_{i<k}X_i \to X_k)$ in order to simulate IPS over T using ips.

THEOREM 4.5 (HA$^\omega$). IPS $=_T$ ips.

PROOF. It is clear that IPS is a generalisation of ips. We now show that IPS is T-definable from ips, following the same ideas used to show that EPS is T-equivalent to eps (Section 3.2). In fact, the proof here is slightly simpler since we do not have to worry about the length function l. Let $\tilde{\varepsilon}_k$ be as defined in Lemma 3.10. The infinite (simple) product of selection functions applied to $\tilde{\varepsilon}$ gives

$$\mathsf{ips}_0(\tilde{\varepsilon}) \quad : \quad J(\Pi_i(\Pi_{k<i}X_k \to X_i)).$$

IPS can then be defined as, where $s\colon \Pi_{k<i}X_k$ so that $|s| = i$,

$$\mathsf{IPS}_s(\varepsilon)(q^{\Pi_j X_{|s|+j}\to R}) \overset{\Pi_j X_{|s|+j}}{=} (\mathsf{ips}_{|s|}(\tilde{\varepsilon})(q^{[s]}))^{[s]}$$

where $\alpha^{[s]}(i) \overset{X_i}{=} \alpha(i)(s * [\alpha^{[s]}](i))$ and $q^{[s]}(\alpha) = q(\alpha^{[s]})$. Unfolding the definitions

$$\begin{aligned}
\mathsf{IPS}_s(\varepsilon)(q)(i) &\overset{X_{|s|+i}}{=} (\mathsf{ips}_{|s|}(\tilde{\varepsilon})(q^{[s]}))^{[s]}(i)\\
&= \mathsf{ips}_{|s|}(\tilde{\varepsilon})(q^{[s]})(i)(s*t)\\
&\overset{\text{D4.4}}{=} \tilde{\varepsilon}_{|s|+i}(\lambda f.(q^{[s]})_{u*f}(\mathsf{ips}_{|s|+i+1}(\tilde{\varepsilon})((q^{[s]})_{u*f})))(s*t)\\
&= \varepsilon_{s*t}(\lambda x.(q^{[s]})_{u*\lambda r.x}(\mathsf{ips}_{|s|+i+1}(\tilde{\varepsilon})((q^{[s]})_{u*\lambda r.x})))\\
&\overset{(i,ii)}{=} \varepsilon_{s*t}(\lambda x.q_{t*x}(\mathsf{ips}_{|s|+i+1}(\tilde{\varepsilon})((q^{[s]})_{u*\lambda r.x}))^{[s*t*x]})\\
&\overset{(iii)}{=} \varepsilon_{s*t}(\lambda x.q_{t*x}(\mathsf{ips}_{|s|+i+1}(\tilde{\varepsilon})((q_{t*x})^{[s*t*x]}))^{[s*t*x]})\\
&= \varepsilon_{s*t}(\lambda x.q_{t*x}(\mathsf{IPS}_{s*t*x}(\varepsilon)(q_{t*x})))\\
&= \varepsilon_{s*t}(\lambda x.q_{t*x}(\overline{\mathsf{IPS}_{s*t*x}(\varepsilon)}(q_{t*x}))
\end{aligned}$$

where $t = [(\mathsf{ips}_{|s|}(\tilde{\varepsilon})(q^{[s]}))^{[s]}](i)$ and $u = [\mathsf{ips}_{|s|}(\tilde{\varepsilon})(q^{[s]})](i)$. We used that

(i) $(t*\alpha)^{[s]} = t^{[s]} * \alpha^{[s*t^s]}$, for all t, s, α,

(ii) $u^{[s]} = t$, for t, s and u as above

(iii) $(q^{[s]})_{u*\lambda r.x} = (q_{t*x})^{[s*t*x]}$.

⊣

REMARK 4.6. *Notice that an implicitly controlled product of quantifiers* IPQ

$$\mathsf{IPQ}_s(\phi) = \phi_s \otimes_d \lambda x.\mathsf{IPQ}_{s*x}(\phi) \tag{10}$$

does not exist, in the sense that the equation above is inconsistent. It is enough to consider the case when $R = \mathbb{B}$ and $X_i = \mathbb{B}$. Take $\phi_s(p) = \neg p(\mathsf{true})$. Moreover, take q to be any constant function. Then, we have

$$\begin{aligned}
\mathsf{IPQ}_{\langle\rangle}(\phi)(q) &= \phi_{\langle\rangle}(\lambda x.\mathsf{IPQ}_{\langle x\rangle}(\phi)(q_x))\\
&= \neg\mathsf{IPQ}_{\langle\mathsf{true}\rangle}(\phi)(q_{\mathsf{true}})\\
&= \neg\mathsf{IPQ}_{\langle\mathsf{true}\rangle}(\phi)(q)\\
&= \neg\mathsf{IPQ}_{\langle\rangle}(\phi)(q),
\end{aligned}$$

as $q = q_{\mathsf{true}}$ since q is constant, and $\mathsf{IPQ}_{\langle x\rangle}(\phi) = \mathsf{IPQ}_{\langle\rangle}(\phi)$ since the quantifiers ϕ_s do not depend on s. See also section 5.6 of [11] where a different argument is given in the specific case of the model of continuous functionals.

4.3. Relation to modified bar recursion. The proof that ips interprets full classical analysis, via modified realizability, is very similar to the one given in [4, 5] that *modified bar recursion* MBR interprets full classical analysis. In this section we show how MBR corresponds directly to the infinite iteration of a different form of binary product of selection functions. We also show (Sections 5.2 and 5.3) that this different product when iterated leads to a form of bar recursion (MBR) which is nevertheless T-equivalent to IPS.

DEFINITION 4.7. *Given a function $\varepsilon \in (X \to R) \to X \times Y$ and a selection function $\delta \in JY$ define a selection function $\varepsilon \tilde{\otimes} \delta \in J(X \times Y)$ as*

$$(\varepsilon \tilde{\otimes} \delta)(p) \overset{X\times Y}{=} \varepsilon(\lambda x.p(x, b(x)))$$

where $b(x) \overset{Y}{=} \delta(\lambda y.p(x, y))$. We shall also consider a dependent version $\tilde{\otimes}_d$ of the product where $\delta\colon X \to JY$ and $b(x) = \delta(x, \lambda y.p(x, y))$.

The above construction shows how a mapping of type $(X \to R) \to X \times Y$ can be extended to a selection function on the product space, given a selection function on Y. We shall use this with $X = X_n$ and $Y = \Pi_i X_{i+n+1}$, so that we obtain a selection function in $J(\Pi_i X_{i+n})$.

DEFINITION 4.8 (mbr). *Let $\varepsilon_n\colon (X_n \to R) \to \Pi_i X_{n+i}$ and $\varepsilon = (\varepsilon_n)_{n\in\mathbb{N}}$. Define the* iterated skewed product mbr *as*

$$\mathsf{mbr}_n(\varepsilon) \overset{J(\Pi_i X_{n+i})}{=} \varepsilon_n \tilde{\otimes} \mathsf{mbr}_{n+1}(\varepsilon).$$

Unfolding the definition of $\tilde{\otimes}$ we have

$$\mathsf{mbr}_n(\varepsilon)(q) \overset{\Pi_i X_{n+i}}{=} \varepsilon_n(\lambda x.q_x(\mathsf{mbr}_{n+1}(\varepsilon)(q_x)). \tag{11}$$

Define also the dependent iterated skewed product MBR

$$\mathsf{MBR}_s(\varepsilon) \overset{J(\Pi_i X_{|s|+i})}{=} \varepsilon_s \tilde{\otimes}_d (\lambda x.\mathsf{MBR}_{s*x}(\varepsilon)), \tag{12}$$

where in this case $\varepsilon_s\colon (X_{|s|} \to R) \to \Pi_i X_{|s|+i}$. We name this mbr *and* MBR *because we will show this is essentially* modified bar recursion *as defined in [4, 5].*

We think of ε as a sequence of *skewed selection functions*. The idea is that sometimes a witness for X_k is automatically a witness for all types X_i for $i \geqslant k$. In such cases, a selection function $\varepsilon_n\colon (X_n \to R) \to X_n$ gives rise to a skewed selection function $\varepsilon_n\colon (X_n \to R) \to \Pi_{i\geqslant n} X_i$, so that the more intricate product of selection functions (Definition 2.4) can be replaced by the simpler product given in Definition 4.7.

As with IPS and EPS (Sections 3.2 and 4.2), we now show that also the *simple* iterated skewed product mbr (cf. Section 4.3) is T-equivalent to its *dependent* variant MBR.

THEOREM 4.9 (HA^ω). $\mathsf{mbr} \geqslant_T \mathsf{MBR}$.

PROOF. Given a sequence of types X_i let

$$Y_j \equiv \mathbb{B} \times (\Pi_{i<j} X_i \to \Pi_k X_{j+k}).$$

In order to define MBR of type (X_i, R) we use mbr of type (Y_j, R). The intuition for the construction below is the same as the one used to show that eps T-defines EPS (Theorem 3.10), except that here we need an extra boolean flag as the whole result of the skewed selection function will be returned on the first position of the output. The flag is used so that functions quering such sequences can know which are proper values and which are dummy values.
For any given $\varepsilon_s \colon (X_j \to R) \to \Pi_k X_{j+k}$, where $s \colon \Pi_{i<j} X_i$, define

$$\tilde{\varepsilon}_j \colon (Y_j \to R) \to \Pi_k Y_{j+k}$$

by

$$\tilde{\varepsilon}_j(P^{Y_j \to R})(k) \overset{Y_{j+k}}{=} \begin{cases} \langle \mathsf{tt}, \lambda t^{\Pi_{i<j} X_i}.\varepsilon_t(\lambda x^{X_j}.P(\hat{x}))\rangle & \text{if } k = 0, \\ \langle \mathsf{ff}, \mathbf{0}^{\Pi_{i<j+k} X_i \to \Pi_i X_{j+k+i}}\rangle & \text{if } k > 0, \end{cases}$$

where $\hat{x} = \langle \mathsf{tt}, \lambda s^{\Pi_{i<j} X_i}.\langle x^{X_j}, \mathbf{0}^{X_{j+1}}, \mathbf{0}^{X_{j+2}}, \ldots\rangle\rangle$. The simple skewed product of $\tilde{\varepsilon}$ gives $\mathsf{mbr}_0(\tilde{\varepsilon}) \colon J(\Pi_i Y_i)$. Hence, MBR can then be defined from mbr as

$$\mathsf{MBR}_s(\varepsilon)(q^{\Pi_i X_{|s|+i} \to R}) \overset{\Pi_i X_{|s|+i}}{=} (\mathsf{mbr}_{|s|}(\tilde{\varepsilon})(q^{[s]}))^{[s]}.$$

Here, for $\alpha \colon \Pi_i Y_{|s|+i}$, we are defining

$$\alpha^{[s]}(i) \overset{X_{|s|+i}}{=} \begin{cases} f(s * [\alpha^{[s]}](i))(0) & \text{if } \alpha(i) = \langle \mathsf{tt}, f^{\Pi_{j<|s|+i} X_j \to \Pi_k X_{|s|+i+k}}\rangle, \\ g(s * [\alpha^{[s]}](n))(i-n) & \text{otherwise,} \end{cases}$$

where n is the the greatest $n < i$ such that $\alpha(n)$ is of the form $\langle \mathsf{tt}, g\rangle$ (or $n = i$ if such does not exist). Note that $\alpha^{[s]} \colon \Pi_i X_{|s|+i}$. Also, for $q \colon \Pi_i X_{|s|+i} \to R$ we define

$$q^{[s]}(\alpha) \overset{R}{=} q(\alpha^{[s]})$$

so $q^{[s]} \colon \Pi_i Y_{|s|+i} \to R$. Unfolding the definitions we have

$$\begin{aligned} \mathsf{MBR}_s(\varepsilon)(q) &\overset{\Pi_i X_{|s|+i}}{=} (\mathsf{mbr}_{|s|}(\tilde{\varepsilon})(q^{[s]}))^{[s]} \\ &\overset{(11)}{=} \big(\tilde{\varepsilon}_{|s|}(\lambda f^{Y_{|s|}}.(q^{[s]})_f(\mathsf{mbr}_{|s|+1}(\tilde{\varepsilon})((q^{[s]})_f)))\big)^{[s]} \\ &\overset{(i)}{=} \varepsilon_s(\lambda x^{X_{|s|}}.(q^{[s]})_{\hat{x}}(\mathsf{mbr}_{|s|+1}(\tilde{\varepsilon})((q^{[s]})_{\hat{x}}))) \\ &\overset{(ii)}{=} \varepsilon_s(\lambda x^{X_{|s|}}.q_x(\big(\mathsf{mbr}_{|s*x|}(\tilde{\varepsilon})((q_x)^{[s*x]})\big)^{[s*x]})) \\ &= \varepsilon_s(\lambda x^{X_{|s|}}.q_x(\mathsf{MBR}_{s*x}(\varepsilon)(q_x))), \end{aligned}$$

using

(*i*) $(\tilde{\varepsilon}_{|s|}(F))^{[s]} = \varepsilon_s(\lambda x.F(\hat{x}))$
(*ii*) $(q^{[s]})_{\hat{x}}(\alpha) = q((\hat{x} * \alpha)^{[s]}) = q_x(\alpha^{[s*x]}) = (q_x)^{[s*x]}(\alpha)$

which can be shown directly from the definitions of $q^{[s]}$, $\alpha^{[s]}$ and $\hat{x}$. ⊣

We now show that a slight generalisation of modified bar recursion [4, 5] is T-equivalent to the iterated product of skewed selection functions.

THEOREM 4.10 ($\mathrm{HA}^\omega + \mathrm{BI} + \mathrm{CONT}$). *Define* MBR' *as*

$$\mathsf{MBR}'_s(\varepsilon)(q) \overset{\Pi_i X_i}{=} s * \varepsilon_s(\lambda x^{X_{|s|}}.q(\mathsf{MBR}'_{s*x}(\varepsilon)(q))) \tag{13}$$

where $q\colon \Pi_i X_i \to R$ *and* $\varepsilon_s\colon (X_{|s|} \to R) \to \Pi_i X_{|s|+i}$. *Then* MBR *and* MBR′ *are T-equivalent.*

PROOF. MBR′ is a generalisation of modified bar recursion (as defined in [4, 5]) to sequence types. If all $X_i = X$ we have precisely the definition given in [4, 5]. For one direction, let $q\colon \Pi_i X_i \to R$ and $s\colon \Pi_{i<n} X_i$ and define

$$\mathsf{MBR}'_s(\varepsilon)(q) \overset{\Pi_i X_i}{=} s * \mathsf{MBR}_s(\varepsilon)(q_s).$$

Unfolding definitions we have

$$\begin{aligned}
\mathsf{MBR}'_s(\varepsilon)(q) &= s * \mathsf{MBR}_s(\varepsilon)(q_s)\\
&\overset{(12)}{=} s * ((\varepsilon_s \tilde{\otimes}_d \lambda x.\mathsf{MBR}_{s*x}(\varepsilon))(q_s))\\
&\overset{(D4.7)}{=} s * \varepsilon_s(\lambda x^{X_{|s|}}.q_{s*x}(\mathsf{MBR}_{s*x}(\varepsilon)(q_{s*x})))\\
&= s * \varepsilon_s(\lambda x^{X_{|s|}}.q(s * x * \mathsf{MBR}_{s*x}(\varepsilon)(q_{s*x})))\\
&= s * \varepsilon_s(\lambda x^{X_{|s|}}.q(\mathsf{MBR}'_{s*x}(\varepsilon)(q))).
\end{aligned}$$

For the other direction, let $q\colon \Pi_i X_{|s|+i} \to R$. Define

$$\mathsf{MBR}_s(\varepsilon)(q) \overset{\Pi_i X_{|s|+i}}{=} \mathsf{MBR}'_{\langle\rangle}(\lambda t.\varepsilon_{s*t})(q).$$

We then have

$$\begin{aligned}
\mathsf{MBR}_s(\varepsilon)(q) &= \mathsf{MBR}'_{\langle\rangle}(\lambda t.\varepsilon_{s*t})(q)\\
&\overset{(13)}{=} \varepsilon_s(\lambda x^{X_{|s|}}.q(\mathsf{MBR}'_x(\lambda t.\varepsilon_{s*t})(q)))\\
&\overset{(+)}{=} \varepsilon_s(\lambda x^{X_{|s|}}.q_x(\mathsf{MBR}'_{\langle\rangle}(\lambda t.\varepsilon_{s*x*t})(q_x)))\\
&= \varepsilon_s(\lambda x^{X_{|s|}}.q_x(\mathsf{MBR}_{s*x}(\varepsilon)(q_x))),
\end{aligned}$$

where

$$(+)\quad \mathsf{MBR}'_{x*r}(\lambda t.\varepsilon_{s*t})(q) = x * \mathsf{MBR}'_r(\lambda t.\varepsilon_{s*x*t})(q_x)$$

can be proven by bar induction on the sequence r, assuming continuity of q (similar to Theorem 3.5). ⊣

COROLLARY 4.11. *Gandy's functional* Γ *is T-equivalent to* MBR *with* $X_i = \mathbb{N}$ *for all* $i \in \mathbb{N}$.

PROOF. It has been shown in [5] that the Γ functional is T-equivalent to MBR of lowest type. It remains to observe that the equivalence of Theorem 4.10 respects the types. ⊣

QUESTION 4.12. *It should be mentioned that in [2] yet another form of bar recursion is used for the interpretation of the double negation shift (although they also use modified bar recursion when dealing with dependent choice). We refer to this different bar recursion as the* bbc *functional. It completely open how the* bbc *functional fits into the picture in Figure 1. For more information on the* bbc *functional see [3].*

§**5. Further Inter-definability Results.** In this section we prove three further inter-definability results, namely IPS ⩾ EPQ, ips ⩾ mbr and MBR ⩾ IPS.

5.1. IPS $\geqslant$ EPQ**.** It has been shown in [5] that BR is T-definable from modified bar recursion. Here we simplify that construction and use it to show that EPQ is T-definable from IPS. Moreover, we make explicit the assumption in [5] that for all $\omega\colon \Pi_i X_i \to \mathbb{N}$ and $\alpha\colon \Pi_i X_i$ there exists an n such that $\omega(\overline{\alpha,n}) < n$, i.e.

$$\forall\omega\forall\alpha\exists n(\omega(\overline{\alpha,n}) < n). \tag{14}$$

First we prove that (the totalisation of) *Spector's search functional* is definable in Gödel's system T.

LEMMA 5.1 (HA$^\omega$). *The totalisation of* Spector's search functional

$$\mu_{\mathsf{sc}}(\omega)(\alpha) = \mathsf{least}\, n(\omega(\overline{\alpha,n}) < n)$$

is T-definable. More precisely, there exists a term t in Gödel's system T such that the following is provable in HA$^\omega$

$$\exists n(\omega(\overline{\alpha,n}) < n) \to (\omega(\overline{\alpha, t\omega\alpha}) < t\omega\alpha \wedge \forall i < t\omega\alpha(\omega(\overline{\alpha,i}) \geqslant i)).$$

In particular, HA$^\omega$ + (14) $\vdash \omega(\overline{\alpha,t\omega\alpha}) < t\omega\alpha$.

PROOF. We show how the unbounded search in μ_{sc} can be turned into a bounded search. Abbreviate $A_n(\omega,\alpha) = (\omega(\overline{\alpha,n}) < n)$. Consider the following construction, given $\alpha\colon \Pi_i X_i$ define $\alpha^\omega\colon \Pi_i X_i$ as

$$\alpha^\omega(i) = \begin{cases} \mathbf{0}^{X_i} & \text{if } \exists k \leqslant i+1\, A_k(\omega,\alpha) \\ \alpha(i) & \text{otherwise.} \end{cases}$$

Assume $\exists n(\omega(\overline{\alpha,n}) < n)$. Let n is the least number such that $A_n(\omega,\alpha)$ holds. Then it is easy to see that $\alpha^\omega = \overline{\alpha, n-1}$. Because n is least, we must have that $\omega(\alpha^\omega) \geqslant n-1$, and hence $n \leqslant \omega(\alpha^\omega)+1$. Therefore, $\omega(\alpha^\omega)+1$ serves as an upper bound on the search μ_{sc}, i.e. $t\omega\alpha = \mu n < \omega(\alpha^\omega)+1(\omega(\overline{\alpha,n}) < n)$. ⊣

The construction above shows that Spector's search functional can be made total in system T, so that whenever it is well-defined for inputs ω and α the term t indeed computes the right value.

REMARK 5.2. *In particular we have that bar recursion* BR *as formulated in Section 3.3 is T-equivalent to the version* $\widetilde{\mathsf{BR}}$ *where we slightly change the stopping condition to guarantee monotonicity, as used in [12],*

$$\widetilde{\mathsf{BR}}^\omega_s(\phi)(q) \stackrel{R}{=} \begin{cases} q_t(\mathbf{0}) & \text{if } \exists t \preceq s(\omega_t(\mathbf{0}) < |t|) \\ \phi_s(\lambda x^{X_{|s|}}.\widetilde{\mathsf{BR}}^\omega_{s*x}(\phi)(q)) & \text{otherwise,} \end{cases}$$

where t (in $q_t(\mathbf{0})$) is the shortest t such that $\omega(\hat{t}) < |t|$. Indeed, given ω, we can define

$$\tilde{\omega}(\alpha) = \mu_{\mathsf{sc}}(\omega)(\alpha)$$

so that once $\tilde{\omega}_s(\mathbf{0}) < |s|$ holds for some sequence s then it also holds for all extensions of s. The same holds for SBR*.*

THEOREM 5.3 (HA$^\omega$ + (14)). IPS $\geqslant_T$ EPQ.

PROOF. We use a generalisation of the search operator above, namely

$$\mu^k_{\mathsf{sc}}(\omega)(\alpha) = \mathsf{least}\, n(\omega(\overline{\alpha,n}) < k+n),$$

which is clearly definable in μ_{sc}. Let $\psi_s \colon K_R X_{|s|}$ be a given family of quantifiers. Let also $X \uplus Y$ denote the sum of types X and Y, which can be implemented as $\mathbb{B} \times X \times Y$, since we assume all types are inhabited. Let $\mathsf{inj}_X \colon X \to X \uplus Y$ and $\mathsf{inj}_Y \colon X \uplus Y$ be the standard injections. We first turn each family of quantifiers $\psi_s \colon K_R X_{|s|}$, where $s \colon \Pi_{i<|s|} X_i$, into a family of selection functions $\tilde{\psi}_t$ of type $J_R(X_{|t|} \uplus R)$, where $t \colon \Pi_{i<|t|}(X_i \uplus R)$, as

$$\tilde{\psi}_t(F^{(X_{|t|} \uplus R) \to R}) \overset{X_{|t|} \uplus R}{=} \mathsf{inj}_R(\psi_{\check{t}}(\lambda x^{X_{|t|}}.F(\mathsf{inj}_{X_{|t|}} x)))$$

where $\check{(\cdot)} \colon \Pi_{i<n}(X_i \uplus R) \to \Pi_{i<n} X_i$ is defined as

$$(\check{s})_i \overset{X_i}{=} \begin{cases} x_i & \text{if } s_i = \mathsf{inj}_{X_i}(x_i) \\ \mathbf{0}^{X_i} & \text{otherwise.} \end{cases}$$

We will also make use of the dual operation that given an $s \colon \Pi_{i<n} X_i$ injects this into $\tilde{s} \colon \Pi_{i<n}(X_i \uplus R)$. Finally, given $s \colon \Pi_{i<k} X_i$ and $q \colon \Pi_i X_{k+i} \to R$ and $\omega \colon \Pi_i X_i \to \mathbb{N}$ define $q^{\omega,s} \colon \Pi_i(X_{k+i} \uplus R) \to R$ as

$$q^{\omega,s}(\alpha^{\Pi_i(X_{k+i} \uplus R)}) \overset{R}{=} \begin{cases} q([\check{\alpha}](N) * \mathbf{0}) & \text{if } \forall i < N\,(\alpha(i) \in X_{k+i}) \\ a & \text{otherwise,} \end{cases}$$

where $N = \mu^{|s|}_{\mathsf{sc}}(\omega_s)(\check{\alpha})$ and $\alpha(\mu i < N\,(\alpha(i) \in R)) = \mathsf{inj}_R(a)$. By Lemma 5.1, under the assumption (14), $q^{\omega,s}$ is T-definable. Intuitively, when $q^{\omega,s}$ reads an input sequence α it uses Spector's search functional on ω to compute the point N where Spector's condition is satisfied in $\check{\alpha}$. If all values in α up to that point are X_i values, then we apply q. Otherwise, some value is encoding the return of the computation R, and the first such value is then returned. We claim that

$$\mathsf{EPQ}^{\omega}_s(\psi)(q^{\Pi_i X_{|s|+i} \to R}) \overset{R}{=} q^{\omega,s}(\mathsf{IPS}_{\tilde{s}}(\tilde{\psi})(q^{\omega,s}))$$

does the job. Let us unfold the definitions. We consider two cases.

Assume first that $\omega(\hat{s}) < |s|$. Then

$$\begin{aligned} \mathsf{EPQ}^{\omega}_s(\psi)(q) &\overset{R}{=} q^{\omega,s}(\mathsf{IPS}_{\tilde{s}}(\tilde{\psi})(q^{\omega,s})) \\ &= q(\mathbf{0}), \end{aligned}$$

since in this case $N = \mu^{|s|}_{\mathsf{sc}}(\omega_s)(\check{\alpha}) = 0$.

On the other hand, if $\omega_s(\mathbf{0}) \geqslant |s|$, we have

$$\begin{aligned} \mathsf{EPQ}^{\omega}_s(\psi)(q) &\overset{R}{=} q^{\omega,s}(\mathsf{IPS}_{\tilde{s}}(\tilde{\psi})(q^{\omega,s})) \\ &\overset{(\text{D4.4})}{=} q^{\omega,s}(\tilde{\psi}_s(\lambda u^{X_{|s|} \uplus R}.(q^{\omega,s})_u(\mathsf{IPS}_{\tilde{s}}(\tilde{\psi})((q^{\omega,s})_u)) * \dots) \\ &= \psi_s(\lambda x.(q^{\omega,s})_{\mathsf{inj}_{X_{|s|}}(x)}(\mathsf{IPS}_{\tilde{s} * \mathsf{inj}_{X_{|s|}}(x)}(\tilde{\psi})((q^{\omega,s})_{\mathsf{inj}_{X_{|s|}}(x)}))) \\ &\overset{(i,ii)}{=} \psi_s(\lambda x.(q_x)^{\omega,s*x}(\mathsf{IPS}_{\widetilde{s*x}}(\tilde{\psi})((q_x)^{\omega,s*x}))) \\ &= \psi_s(\lambda x.\mathsf{EPQ}^{\omega}_{s*x}(\psi)(q_x)) \\ &= (\psi_s \otimes_d \lambda x.\mathsf{EPQ}^{\omega}_{s*x}(\psi))(q), \end{aligned}$$

noticing that (i) $(q^{\omega,s})_{\mathsf{inj}_{X_{|s|}}(x)}(\alpha) = (q_x)^{\omega,s*x}(\alpha)$ and (ii) $\tilde{s} * \mathsf{inj}_{X_{|s|}}(x) = \widetilde{s * x}$. $\dashv$

REMARK 5.4. *As shown in [14, 15], if one extends system T with Spector's bar recursion, one can actually prove (14). Hence, the result above says that in all models of system T where* EPQ *could exist, it indeed does whenever* IPS *also exists. We leave it as an open question whether* IPS *already defines* EPQ *without assuming (14).*

5.2. $\mathsf{ips} \geqslant \mathsf{mbr}$**.** We now show that MBR and IPS are T-equivalent. In this section we show that MBR is T-definable from IPS. The converse will be shown in the following section. Given that mbr T-defines MBR, it is enough to show that $\mathsf{ips} \geqslant_T \mathsf{mbr}$

THEOREM 5.5 (HA^ω). $\mathsf{ips} \geqslant_T \mathsf{mbr}$.

PROOF. Given a type X let us denote by X' the type $\mathbb{B} \times X$. In order to define mbr of type (X_i, R) we use ips of type (X_i', R). The main idea for the construction is that we can turn a skewed selection function into a proper selection functions as follows. Given skewed selection functions $\varepsilon_i \colon (X_i \to R) \to \Pi_j X_{i+j}$ we define selection functions $\tilde{\varepsilon}_i \colon J(\Pi_j X'_{i+j})$ as

$$\tilde{\varepsilon}_i(f^{\Pi_j X'_{i+j} \to R}) \overset{\Pi_j X'_{i+j}}{=} \lambda j.\langle \mathsf{ff}, \varepsilon_i(\lambda x^{X_i}.f(\hat{x}))(j)\rangle,$$

where

$$\hat{x}(j) = \begin{cases} \langle \mathsf{tt}, x^{X_i}\rangle & \text{if } j = 0 \\ \langle \mathsf{tt}, \mathbf{0}^{X_{i+j}}\rangle & \text{if } j > 0. \end{cases}$$

Intuitively, the booleans $\{\mathsf{tt}, \mathsf{ff}\}$ are used to distinguish between values returned by ε_i and those values $\hat{x}$ passed into a recursive call. Hence, given a $q \colon \Pi_j X_{i+j} \to R$ we define $\tilde{q} \colon \Pi_k(\Pi_j X'_{i+k+j}) \to R$ as $\tilde{q}(\alpha) = q(\tilde{\alpha})$ where, given $\alpha \colon \Pi_k(\Pi_j X'_{i+k+j})$ we define $\tilde{\alpha} \colon \Pi_j X_{i+j}$ as

$$\tilde{\alpha}(j) \overset{X_{i+j}}{=} \begin{cases} (\alpha(j)(0))_1 & \text{if } \forall k < j\, ((\alpha(k)(0))_0 = \mathsf{tt} \\ (\alpha(k)(j-k))_1 & \text{otherwise,} \end{cases}$$

where $k = \mu k < j\, (\alpha(k)(0))_0 = \mathsf{ff}$. The construction $\tilde{\alpha}$ receives as input a matrix $\alpha \colon \Pi_i \Pi_{j \geqslant i} X_j$ and produces a sequence $\Pi_j X_j$ as follows: As long as the value of $\alpha(j)(0)$ is some $\hat{x}$ (boolean will be tt) we filter out the x; once we reach a value returned by an ε_k (boolean will be ff) then we return the whole sequence returned by the skewed selection function ε_k. We claim that mbr can be defined as

$$\mathsf{mbr}_i(\varepsilon)(q) \overset{\Pi_j X_{i+j}}{=} \left(\left(\mathsf{ips}_i(\tilde{\varepsilon})(\tilde{q})\right)^{\Pi_k \Pi_j X'_{i+k+j}}(0)\right)^1$$

where $\varepsilon_i \colon (X_i \to R) \to \Pi_j X_{i+j}$ and $q \colon \Pi_j X_{i+j} \to R$. Recall that given a sequence $\beta \colon \Pi_{i<n}(X_i \times Y_i)$ we write $\beta^1 \colon \Pi_{i<n} Y_i$ for the projection of the sequence on the second coordinates. Unfolding definitions

$$\begin{aligned}
\mathsf{mbr}_i(\varepsilon)(q) &\overset{\Pi_j X_{i+j}}{=} (\mathsf{ips}_i(\tilde{\varepsilon})(\tilde{q})(0))^1 \\
&\overset{(\text{L4.2})}{=} (\tilde{\varepsilon}_i(\lambda\alpha^{\Pi_j X'_{i+j}}.\tilde{q}_\alpha(\mathsf{ips}_{i+1}(\tilde{\varepsilon})(\tilde{q}_\alpha))))^1 \\
&= \varepsilon_i(\lambda x^{X_i}.\tilde{q}_{\hat{x}}(\mathsf{ips}_{i+1}(\tilde{\varepsilon})(\tilde{q}_{\hat{x}}))) \\
&\overset{(i)}{=} \varepsilon_i(\lambda x^{X_i}.\widetilde{(q_x)}(\mathsf{ips}_{i+1}(\tilde{\varepsilon})(\widetilde{(q_x)}))) \\
&\overset{(ii)}{=} \varepsilon_i(\lambda x^{X_i}.q_x((\mathsf{ips}_{i+1}(\tilde{\varepsilon})(\widetilde{(q_x)}))(0))^1)) \\
&= \varepsilon_i(\lambda x^{X_i}.q_x(\mathsf{mbr}_{i+1}(\varepsilon)(q_x))) \\
&= (\varepsilon_i \mathbin{\tilde{\otimes}} \mathsf{mbr}_{i+1}(\varepsilon))(q),
\end{aligned}$$

using that

(*i*) $\tilde{q}_{\hat{x}}(\beta) = \widetilde{(q_x)}(\beta)$.
(*ii*) $\tilde{\beta} = (\mathsf{ips}_{i+1}(\tilde{\varepsilon})(\widetilde{(q_x)}))(0))^1$, for $\beta = \mathsf{ips}_{i+1}(\tilde{\varepsilon})(\widetilde{(q_x)})$.

⊣

5.3. MBR $\geqslant$ IPS. We now show that the implicitly controlled product of selection functions IPS is T-definable from (and hence T-equivalent to) modified bar recursion MBR.

THEOREM 5.6 (HA^ω). $\mathsf{MBR} \geqslant_T \mathsf{IPS}$.

PROOF. Let $Y_n = \Sigma_i \Pi_{n \leqslant j \leqslant i} X_j$. We will show that IPS of type (X_i, R) is T-definable from MBR of type (Y_i, R). We make use of the following two mappings: $G\colon X_i \to Y_i$ is defined as

$$G(x^{X_i}) \overset{Y_i}{=} \langle x \rangle$$

and $F\colon \Pi_{i \geqslant n} Y_i \to \Pi_{i \geqslant n} X_i$

$$F(\alpha)(i) \overset{X_i}{=} \alpha(j)(i)$$

where $j \leqslant i$ is the first position such that $i < j + |\alpha(j)|$. One can think of $F(\alpha)$ as concatenating a given sequence of finite sequences $\alpha(i)$. We will also apply G to sequences, with the understanding that this means applying G pointwise. Given selection functions $\varepsilon_s\colon J_R X_{|s|}$ define, by course-of-values, skewed selection functions of type

$$\nu_r\colon (Y_{|r|} \to R) \to \Pi_i Y_{|r|+i},$$

where $r\colon \Pi_{i<|r|} Y_i$, as

$$\nu_r(P^{Y_{|r|} \to R})(i) \overset{Y_{|r|+i}}{=} G(\varepsilon_{F(r*t)}(\lambda x^{X_{|r|+i}}.P(\langle (Ft) * x \rangle)))$$

where $t \overset{\Pi_{j<i} Y_{|r|+j}}{=} [\nu_r(P^{Y_{|r|} \to R})](i)$. Also, given $q\colon \Pi_{i \geqslant n} X_i \to R$ define $\tilde{q}\colon \Pi_{i \geqslant n} Y_i \to R$ as

$$\tilde{q}(\alpha) \overset{R}{=} q(F\alpha).$$

We claim that IPS can be defined from MBR as

$$\mathsf{IPS}_s(\varepsilon)(q) \overset{\Pi_i X_{|s|+i}}{=} F(\mathsf{MBR}_{Gs}(\nu)(\tilde{q})),$$

which can be shown as follows:

$$\begin{array}{lcl}
\mathsf{IPS}_s(\varepsilon)(q)(i) & \overset{X_{|s|+i}}{=} & F(\mathsf{MBR}_{Gs}(\nu)(\tilde{q}))(i)\\
& = & F(\nu_{Gs}(\lambda y^{Y_{|s|}}.\tilde{q}_y(\mathsf{MBR}_{(Gs)*y}(\nu)(\tilde{q}_y))))(i)\\
& = & F(\lambda i.G(\varepsilon_{F(G(s)*t)}(\lambda x.\tilde{q}_{\langle (Ft)*x\rangle}(\mathsf{MBR}_{(Gs)*\langle (Ft)*x\rangle}(\nu)(\tilde{q}_{\langle (Ft)*x\rangle})))))(i)\\
& \overset{(i)}{=} & \varepsilon_{s*\check{t}}(\lambda x.\tilde{q}_{\langle \check{t}*x\rangle}(\mathsf{MBR}_{(Gs)*\langle \check{t}*x\rangle}(\nu)(\tilde{q}_{\langle \check{t}*x\rangle})))\\
& \overset{(ii)}{=} & \varepsilon_{s*\check{t}}(\lambda x.q_{\check{t}*x}(F(\mathsf{MBR}_{(Gs)*\langle \check{t}*x\rangle}(\nu)(\widetilde{q_{\check{t}*x}}))))\\
& \overset{(iii)}{=} & \varepsilon_{s*\check{t}}(\lambda x.q_{\check{t}*x}(F(\mathsf{MBR}_{G(s*\check{t}*x)}(\nu)(\widetilde{q_{\check{t}*x}}))))\\
& = & \varepsilon_{s*\check{t}}(\lambda x.q_{\check{t}*x}(\mathsf{IPS}_{s*\check{t}*x}(\varepsilon)(q_{\check{t}*x}))),
\end{array}$$

where, for $j < i$, defining $\check{t} = [\mathsf{IPS}_s(\varepsilon)(q)](i)$, we have

$$\begin{array}{lcl}
F(t)(j) & \overset{Y_{|s|+j}}{=} & F([\nu_{Gs}(\lambda y.\tilde{q}_y(\mathsf{MBR}_{(Gs)*y}(\nu)(\tilde{q}_y)))](i))(j)\\
& = & \varepsilon_{F((Gs)*[t](j))}(\lambda x.\tilde{q}_{\langle F([t](j))*x\rangle}(\mathsf{MBR}_{(Gs)*\langle F([t](j))*x\rangle}(\nu)(\tilde{q}_{\langle F([t](j))*x\rangle})))\\
& \overset{(i,ii)}{=} & \varepsilon_{s*[\check{t}](j)}(\lambda x.q_{[\check{t}](j)*x}(\mathsf{IPS}_{s*[\check{t}](j)*x}(\varepsilon)(q_{[\check{t}](j)*x})))\\
& = & (\check{t})(j)
\end{array}$$

using course-of-values induction (induction hypothesis $[F(t)](j) = [\check{t}](j)$). We have also used the following facts which can be easily verified:

(*i*) $F(\lambda i.G(v_i))(i) = v_i$ and $F(G(s) * t) = s * F(t)$

(*ii*) $\tilde{q}_{\langle s\rangle} = \widetilde{q_s}(\alpha)$

(*iii*) $(Gs) * \langle \check{t} * x\rangle = G(s * \check{t} * x)$.

⊣

§**6. Summary of Results.** Figure 1 gives a diagrammatic picture of the inter-definability results presented above. We use a full-line-arrow to represent that the inter-definability holds over HA^ω, whereas a dotted-line-arrow indicates that extra assumptions are needed. We have used extra assumptions in three cases. In Theorem 5.3 we use bar induction, whereas in Theorem 3.5 we have made use of Spector's condition SPEC. It is an interesting open question whether any of these four results can be shown in HA^ω alone.

Acknowledgements. The authors would like to thank Ulrich Berger and Thomas Powell for suggesting some improvements on an earlier version of the paper. The second author also acknowledges support of The Royal Society under grant 516002.K501/RH/kk.

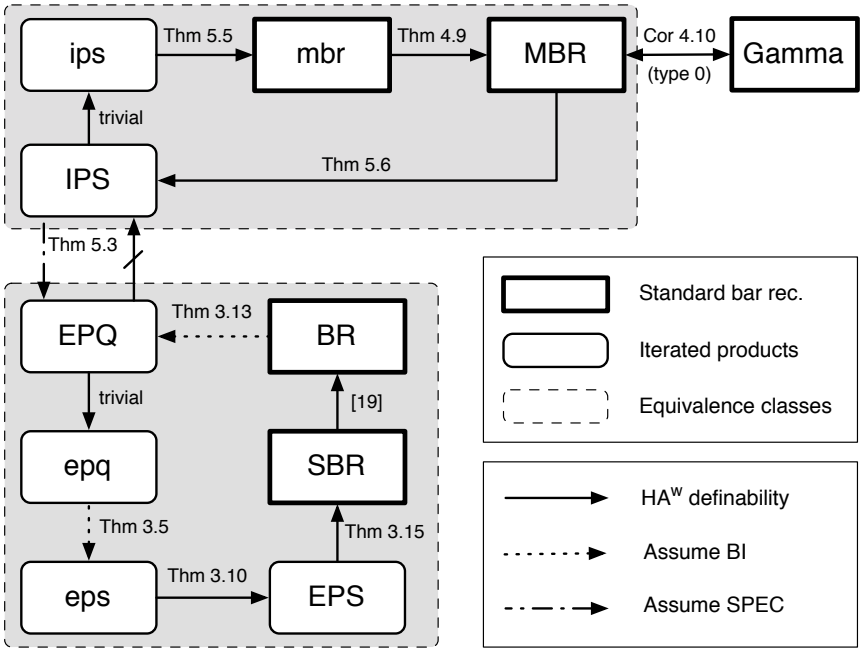

FIGURE 1. Diagram of inter-definability results

REFERENCES

[1] J. AVIGAD and S. FEFERMAN, *Gödel's functional ("Dialectica") interpretation*, ***Handbook of proof theory*** (S. R. Buss, editor), Studies in Logic and the Foundations of Mathematics, vol. 137, North Holland, Amsterdam, 1998, pp. 337–405.

[2] S. BERARDI, M. BEZEM, and T. COQUAND, *On the computational content of the axiom of choice*, ***The Journal of Symbolic Logic***, vol. 63 (1998), no. 2, pp. 600–622.

[3] U. BERGER, *The Berardi-Bezem-Coquand-functional in a domain-theoretic setting*, Draft, July, 2002.

[4] U. BERGER and P. OLIVA, *Modified bar recursion and classical dependent choice*, ***Lecture Notes in Logic***, vol. 20 (2005), pp. 89–107.

[5] ———, *Modified bar recursion*, ***Mathematical Structures in Computer Science***, vol. 16 (2006), pp. 163–183.

[6] U. BERGER and H. SCHWICHTENBERG, *Program extraction from classical proofs*, ***Logic and computational complexity workshop (LCC'94)*** (D. Leivant, editor), Lecture Notes in Computer Science, vol. 960, Springer, Berlin, 1995, pp. 77–97.

[7] M. BEZEM, *Strongly majorizable functionals of finite type: a model for bar recursion containing discontinuous functionals*, ***The Journal of Symbolic Logic***, vol. 50 (1985), pp. 652–660.

[8] M. H. ESCARDÓ, *Exhaustible sets in higher-type computation*, ***Logical Methods in Computer Science***, vol. 4 (2008), no. 3, p. paper 4.

[9] M. H. ESCARDÓ and P. OLIVA, *Computational interpretations of analysis via products of selection functions*, ***Computability in europe 2010, lncs*** (F. Ferreira, B. Lowe, E. Mayordomo, and L. M. Gomes, editors), Springer, 2010, pp. 141–150.

[10] ———, *The Peirce translation and the double negation shift*, ***Programs, Proofs, Processes - CiE 2010, LNCS 6158*** (F. Ferreira, B. Löwe, E. Mayordomo, and L. M. Gomes, editors), Springer, 2010, pp. 151–161.

[11] ———, *Selection functions, bar recursion, and backward induction*, ***Mathematical Structures in Computer Science***, vol. 20 (2010), no. 2, pp. 127–168.

[12] F. FERREIRA and P. ENGRÁCIA, *The bounded functional interpretation of the double negation shift*, ***The Journal of Symbolic Logic***, vol. 75 (2010), no. 2, pp. 759–773.

[13] K. GÖDEL, *Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes*, ***Dialectica***, vol. 12 (1958), pp. 280–287.

[14] W. A. HOWARD, *Functional interpretation of bar induction by bar recursion*, ***Compositio Mathematica***, vol. 20 (1968), pp. 107–124.

[15] U. KOHLENBACH, *Theorie der majorisierbaren und stetigen Funktionale und ihre Anwendung bei der Extraktion von Schranken aus inkonstruktiven Beweisen: Effektive Eindeutigkeitsmodule bei besten Approximationen aus ineffektiven Eindeutigkeitsbeweisen*, ***Ph.D. thesis***, Frankfurt, pp. xxii+278, 1990.

[16] ———, ***Applied proof theory: Proof interpretations and their use in mathematics***, Monographs in Mathematics, Springer, 2008.

[17] D. NORMANN, *The continuous functionals*, ***Handbook of computability theory*** (E. R. Griffor, editor), North Holland, Amsterdam, 1999, pp. 251–275.

[18] P. OLIVA and T. POWELL, *On Spector's bar recursion*, ***Mathematical Logic Quarterly***, vol. 58 (2012), no. 4-5, pp. 356–365.

[19] C. SPECTOR, *Provably recursive functionals of analysis: a consistency proof of analysis by an extension of principles in current intuitionistic mathematics*, ***Recursive function theory: Proc. symposia in pure mathematics*** (F. D. E. Dekker, editor), vol. 5, American Mathematical Society, Providence, Rhode Island, 1962, pp. 1–27.

[20] W. W. TAIT, *Infinitely long terms of transfinite type*, ***Formal Systems and Recursive Functions (Proc. Eighth Logic Colloq., Oxford***, North-Holland, Amsterdam, 1965, pp. 176–185.

[21] A. S. TROELSTRA, ***Metamathematical investigation of intuitionistic arithmetic and analysis***, Lecture Notes in Mathematics, vol. 344, Springer, Berlin, 1973.