The Equivalence of Bar Recursion and Open Recursion*

Thomas Powell
Institut des Hautes Études Scientifiques

July 15, 2014

Abstract

Several extensions of Gödel's system T with new forms of recursion have been designed for the purpose of giving a computational interpretation to classical analysis. One can organise many of these extensions into two groups: those based on *bar recursion*, which include Spector's original bar recursion, modified bar recursion and the more recent products of selections functions, or those based on *open recursion* which in particular include the symmetric Berardi-Bezem-Coquand (BBC) functional. We relate these two groups by showing that both open recursion and the BBC functional are primitive recursively equivalent to a variant of modified bar recursion. Our results, in combination with existing research, essentially complete the classification up to primitive recursive equivalence of those extensions of system T used to give a direct computational interpretation to choice principles.

Keywords. Gödel's system T, primitive recursive equivalence, bar recursion, open recursion.

1 Introduction

In a landmark paper of 1962, C. Spector extended Gödel's Dialectica interpretation of Peano arithmetic to countable choice and hence classical analysis by adding to the primitive recursive functionals a novel form of recursion now know as *bar recursion* [29]. In its broadest sense, bar recursion is a generalisation of primitive recursion to well founded trees, and can be informally described by the scheme

$$B^T(s) = \begin{cases} G(s) & \text{if } s \text{ is a leaf of } T \\ H(s, \lambda x \,.\, B^T(s * x)) & \text{otherwise.} \end{cases}$$

Spector's original bar recursion is just one of the first concrete instances of this kind of recursion, and many new variants have been developed in a number of different contexts. For example, in proof theory novel instances of bar recursion feature in [1] and [5] in order to give new realizability interpretations to countable dependent choice, while in computability theory a form of bar recursion was used in [18] to exhibit a continuous functional with a recursive associate which is nevertheless not S1-S9 computable. More recently, bar recursion in the form of products of selection functions has been shown to have deep connections with game-theory and the computation of so-called generalised Nash equilibria in unbounded sequential games [14, 15]. The relationship between these many variants of bar recursion has been thoroughly investigated in [6, 8, 17, 20, 24].

*The results of this paper form Chapter 12 of the author's PhD dissertation [25]

In 1998, Berardi et al. [1] proposed a beautiful alternative to bar recursion - referred to here as the BBC-functional (or just BBC) - in order to give an efficient computational interpretation to countable choice. Their idea was a symmetric form of recursion, in many ways analogous to bar recursion but in which recursive calls are made over extensions of finite partial functions as opposed to extensions of finite sequences. Despite its apparent simplicity, the behaviour of the BBC-functional is seemingly much harder to understand than bar recursion. An early attempt by Berger [2] to justify BBC in a standard domain-theoretic type structure resorted to a complex non-constructive argument involving Zorn's lemma. A more satisfactory framework was subsequently developed by the same author in [3] where it is shown that BBC is a simple instance of a general schema of open recursion over the lexicographic ordering - a computational analogue of the principle of open induction which in turn forms the contrapositive to the well-known minimal bad sequence argument. Nevertheless, in contrast to bar recursion, relatively little is understood about open recursive functionals, and in particular their relationship to bar recursion remains unknown.

In this article, we prove that both open recursion and even the apparently weaker BBC functional are primitive recursively equivalent to a class of bar recursive functionals that includes the modified bar recursion of [5] and the more recent implicit product of selection functions of [17]. As an immediate consequence we obtain a new proof of the totality of BBC and open recursion, and also verify that neither of these are S1-S9 computable in the total continuous functionals. More importantly, in combination with previous results, we essentially complete the classification of the well-known computational interpretations of analysis according to primitive recursive equivalence. Moreover, we give direct constructions of open recursion and BBC as single instances of bar recursion, and vice-versa, and in doing so hopefully shed some light on the qualitative relationship between these functionals, which may in turn lead to an improved understanding of how they compare in practice when used to extract realizers from proofs.

The organisation of this paper is fairly straightforward. In the remainder of this section we provide some essential preliminary material, before moving onto a brief survey of bar recursion and open recursion in Section 2. Sections 3 and 4 contain our main results: the definability of open recursion from bar recursion and the definability of bar recursion from the BBC-functional, respectively.

1.1 Heyting arithmetic in all finite types

In this paper we study recursion over all finite types. Our formulation of the finite types contains base types $\mathbb{N}$ and $\mathbb{B}$ for natural numbers and booleans, function types $\rho \to \tau$ (which we sometimes denote as τ^ρ), product types $\rho \times \tau$ and finite sequence types ρ^*. We will also make use of a type $\bar{\rho} \equiv \rho + 1$ in order to represent partial sequences as the type $\bar{\rho}^{\mathbb{N}}$ (we assume that it is decidable whether or not a given point is in the domain of a partial sequence). We write $x\colon \rho$ or x^ρ when x has type ρ.

A *discrete* type is any type that can be encoded in $\mathbb{N}$. For example, all of $\mathbb{B}$, $\mathbb{N}$ and $\mathbb{B}\times\mathbb{N}$ are discrete, but $\mathbb{N} \to \mathbb{N}$ is not. The topological significance of the discrete types in the context of the continuous functionals is discussed in [11]. The restriction on some types being discrete will be important later in order to ensure that the defining equations of certain recursors are consistent with Heyting arithmetic.

We work in the standard theory $\mathsf{E\text{-}HA}^\omega$ of fully extensional Heyting arithmetic in all finite types (see e.g. [21, 30] for details), which contains variables and quantifiers for all types, induction for arbitrary formulas, and the usual non-logical constants with their defining axioms, including symbols R_ρ for primitive recursion in all finite-types:

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

There are various ways of formulating equality at higher-types precisely. In [21], $\mathsf{E\text{-}HA}^\omega$ comes equipped

with predicates $=_{\mathbb{N}}$ and $=_{\mathbb{B}}$ for equality between numbers and booleans, while equality for compound types is defined in terms of these. In particular, for function types we have

$$f =_{\rho\to\tau} g :\equiv \forall x^{\rho}(fx =_{\tau} gx).$$

Extensionality of higher-type equality is given via the axioms

$$\forall f^{\rho\to\tau}, x^{\rho}, y^{\rho}(x =_{\rho} y \to fx =_{\tau} fy).$$

The terms of $\mathsf{E\text{-}HA}^{\omega}$ are otherwise known as Gödel's system T, although the *theory* T technically refers to a quantifier-free fragment of $\mathsf{E\text{-}HA}^{\omega}$ (see [30]). The set-theoretic functionals represented by the closed term of $\mathsf{E\text{-}HA}^{\omega}$ are the Gödel primitive recursive functionals of all finite types, and these substantially expand the usual class of primitive recursive functions which allow only primitive recursion of lowest type: in particular the Ackermann function is already definable from $\mathsf{R}_{\mathbb{N}\to\mathbb{N}}$.

The theory $\mathsf{E\text{-}HA}^{\omega}$ allows us to carry out λ-abstraction and definition by cases. Furthermore, quantifier-free formulas in the language of $\mathsf{E\text{-}HA}^{\omega}$ can be given characteristic functions using the recursor, and can therefore be used as clauses in definitions by cases. The symbol $\mathbf{0}_{\rho}$ will denote the canonical zero object of type ρ, while for product types π_i and $\langle\ ,\ \rangle$ denote the usual projection and pairing functions. Given $t\colon \rho_0 \times \rho_1$ we write $t_i := \pi_i t$, and sometimes for sequences $\alpha\colon (\rho_0 \times \rho_1)^{\mathbb{N}}$ we write $\alpha_i := \lambda n.\pi_i \alpha(n)$. For finite sequence types, $|\cdot|$ denotes the length function and $*$ the concatenation of two sequences i.e. $s * t := \langle s_0, \ldots, s_{m-1}, t_0, \ldots, t_{n-1}\rangle$. For a single object $x\colon \rho$ we will write $s * x$ instead of $s * \langle x\rangle$. We also write $s * \alpha$ for the concatenation of s with an infinite sequence α.

For partial sequences $u\colon \bar{\rho}^{\mathbb{N}}$ we write $u(n) = \bot$ for $u(n) = [1]_r$ and $u(n) = x^{\rho}$ for $u(n) = [x]_l$, where $[\cdot]_l$ and $[\cdot]_r$ are the coprojections into $\rho + 1$. We say that u is defined at n whenever $u(n) = x$ for some x^{ρ}, and denote the set of defined values of u as $\mathrm{dom}(u)$. The predicate $n \in \mathrm{dom}(u)$ is decidable. Finally, we will make use of the following basic recursive operations and notational conventions:

- For $\alpha\colon \rho^{\mathbb{N}}$ we define $\mathsf{tail}_n(\alpha) := \lambda k.\alpha(n+k)$, and for $q\colon \rho^{\mathbb{N}} \to \tau$ and $s\colon \rho^*$ we define $q_s := \lambda\alpha\ .\ q(s * \alpha)$.
- The overwrite function for finite sequences $@ : \rho^* \times \rho^{\mathbb{N}} \to \rho^{\mathbb{N}}$ is defined by

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

  and similarly, overwriting partial functions $u\colon \bar{\rho}^{\mathbb{N}}$ is defined by

  $$(u\ @\ \alpha)(n) := \begin{cases} u(n) & \text{if } n \in \mathrm{dom}(u) \\ \alpha(n) & \text{otherwise.} \end{cases}$$

  It will always be clear from the context which of these we mean.
- For $\alpha\colon \rho^{\mathbb{N}}$ we define $[\alpha](n) := \langle \alpha(0), \ldots, \alpha(n-1)\rangle$ and for $s\colon \rho^*$ we define $\widehat{s} := s\ @\ \mathbf{0}_{\rho^{\mathbb{N}}}$. We also define $\overline{\alpha, n} := \widehat{[\alpha](n)}$.
- There is a bounded search operator $\mu\colon \mathbb{B}^{\mathbb{N}} \times \mathbb{N} \to \mathbb{N}$ satisfying

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

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

1.2 Models of E-HA$^\omega$

A standard interpretation of E-HA$^\omega$ is the type-structure $\mathcal{S}^\omega$ of all set theoretic functionals. However, extensions of E-HA$^\omega$ with bar recursion typically only exist in structures that admit some kind of continuity, such as the total continuous functionals $\mathcal{C}^\omega$ of Kleene [19] and Kreisel [22]. In order to keep our paper self-contained and as general as possible we refrain from working in any specific model, but when necessary we will extend E-HA$^\omega$ with the following sequential continuity axiom for discrete types τ, which is valid in $\mathcal{C}^\omega$:

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

Not all models of bar recursive functionals need satisfy Cont, in particular Spector's bar recursion exists in the strongly majorizable functionals $\mathcal{M}^\omega$ ([7]), which only satisfies the weaker condition

$$\forall F^{\rho^{\mathbb{N}}\to\mathbb{N}}, \alpha \exists N(F(\overline{\alpha, N}) < N).$$

However, some level of continuity seems necessary to reason about stronger forms of recursion such as modified bar recursion and open recursion. We leave open the question of whether or not our main results can be verified using a strictly weaker axiom than Cont.

In addition to continuity, we will also require the following schema of relativised bar induction:

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

where P and S are predicates over ρ^* with S quantifier-free, and $\alpha \in S$ and $s \in S$ are shorthand for $\forall n S([\alpha](n))$ and $S(s)$ respectively. The axiom schema of relativised quantifier-free bar induction QF-BI is bar induction for P also restricted to being quantifier-free. In most cases we will not need to relativise the bar induction at all, simply taking $S(s) = \mathsf{true}$.

For a detailed discussion of bar induction in the context of intuitionistic mathematics see [30]. Classically, bar induction follows from dependent choice, and is therefore valid in type-structures of continuous functionals since the type $\rho^{\mathbb{N}}$ is interpreted as the space $\mathbb{N} \to \mathcal{C}_\rho$ of all sequences of continuous objects.

2 Extensions of the Gödel primitive recursive functionals

In this article we consider extensions of system T by functionals $\mathsf{F}_{\underline{\rho}}$ indexed by tuples of types $\underline{\rho}$, defined by recursive equations of the form

$$\mathsf{F}_{\underline{\rho}}(x_0, \ldots, x_{k-1}) = \Omega_{\underline{\rho}}(\mathsf{F}_{\underline{\rho}}, x_0, \ldots, x_{k-1})$$

where $\Omega_{\underline{\rho}}$ is some primitive recursive functional. We make an important distinction between the functional $\mathsf{F}_{\underline{\rho}}$ and its defining axiom, which we label $\mathsf{F}^{\mathsf{ax}}_{\underline{\rho}}[\Phi]$. The latter we consider a formula with one free variable Φ, given by

$$\mathsf{F}^{\mathsf{ax}}_{\underline{\rho}}[\Phi]\ :\ \forall x_0, \ldots, x_{k-1}(\Phi(x_0, \ldots, x_{k-1}) = \Omega_{\underline{\rho}}(\Phi, x_0, \ldots, x_{k-1})),$$

which simply says that the object Φ satisfies the defining equation of $\mathsf{F}_{\underline{\rho}}$. By E-HA$^\omega + \mathsf{F}_{\underline{\rho}}$ we implicitly mean the extension of E-HA$^\omega$ with a new constant $\mathsf{F}_{\underline{\rho}}$ that satisfies $\mathsf{F}^{\mathsf{ax}}_{\underline{\rho}}[\mathsf{F}_{\underline{\rho}}]$, although we do not assume that $\mathsf{F}_{\underline{\rho}}$ is a unique solution to $\mathsf{F}^{\mathsf{ax}}_{\underline{\rho}}[\Phi]$.

Definition 2.1. The hierarchy of functionals $\mathsf{F} \equiv \{\mathsf{F}_{\underline{\rho}}\}_{\underline{\rho} \in X}$ primitive recursively defines the hierarchy $\mathsf{G} \equiv \{\mathsf{G}_{\underline{\sigma}}\}_{\underline{\sigma} \in Y}$ relative to some collection of additional axioms Δ if for each $\underline{\sigma} \in Y$ there exist types $\underline{\rho}_0(\underline{\sigma}), \ldots, \underline{\rho}_{n-1}(\underline{\sigma}) \in X$ and a closed term $t \in \mathsf{E\text{-}HA}^\omega$ such that

$$\mathsf{E\text{-}HA}^\omega + \Delta + \mathsf{F}_{\underline{\rho}_0(\underline{\sigma})} + \ldots + \mathsf{F}_{\underline{\rho}_{n-1}(\underline{\sigma})} \vdash \mathsf{G}^{\mathsf{ax}}_{\underline{\sigma}}[t(\mathsf{F}_{\underline{\rho}_0(\underline{\sigma})}, \ldots, \mathsf{F}_{\underline{\rho}_{n-1}(\underline{\sigma})})].$$

For brevity, we simply write $\mathsf{G} \leq_\mathsf{T} \mathsf{F}$ over Δ, and when Δ is empty, just $\mathsf{G} \leq_\mathsf{T} \mathsf{F}$.

Remark 2.2.

1. We say that $\mathsf{G} \leq_\mathsf{T} \mathsf{F}$ in some model $\mathcal{T}^\omega$ of $\mathsf{E\text{-}HA}^\omega$ whenever $\mathsf{G} \leq_\mathsf{T} \mathsf{F}$ over Δ with $\mathcal{T}^\omega \models \Delta$. For example, if $\Delta \equiv \mathsf{Cont} + \mathsf{QF\text{-}BI}$ then $\mathsf{G} \leq_\mathsf{T} \mathsf{F}$ is valid in the continuous functionals $\mathcal{C}^\omega$.

2. If $\mathsf{G} \leq_\mathsf{T} \mathsf{F}$ and $\mathsf{F} \leq_\mathsf{T} \mathsf{G}$ over Δ then we say that F and G are primitive recursively equivalent ($\mathsf{F} =_\mathsf{T} \mathsf{G}$) over Δ. It is easy to show that the relation $\leq_\mathsf{T}$ is transitive, and therefore $=_\mathsf{T}$ is an equivalence relation on modes of recursion. If $\mathsf{F} =_\mathsf{T} \mathsf{G}$ over Δ then the corresponding extensions of system T can be identified in any model of $\mathsf{E\text{-}HA}^\omega + \Delta$.

Our notion of definability between modes of recursion in all finite types may appear to be quite a weak property in that we impose no restriction on the level of types. If we have proved that $\mathsf{F} \geq \mathsf{G}$ by establishing that $\mathsf{F}_\rho \geq \mathsf{G}_\rho$ for all ρ, we might have just as well done so via the weaker result $\mathsf{F}_{\rho \to \rho} \geq \mathsf{G}_\rho$, for instance. However, as we will see in Section 2.2, primitive recursive definability is indeed a non-trivial and subtle property that yields interesting insights into the relationship between extensions of system T.

Moreover, in all the results that follow, we establish strictly more than definability of the main hierarchies, and succeed in showing that

$$\mathsf{E\text{-}HA}^\omega + \Delta + \mathsf{F}_{\underline{\rho}(\underline{\sigma})} \vdash \mathsf{G}^{\mathsf{ax}}_{\underline{\sigma}}[t(\mathsf{F}_{\underline{\rho}(\underline{\sigma})})],$$

for each $\underline{\sigma}$, in other words defining $\mathsf{G}_{\underline{\sigma}}$ in terms of just one object $\mathsf{F}_{\underline{\rho}(\underline{\sigma})}$ in the hierarchy F, where $\underline{\rho}(\underline{\sigma})$ is uniform in $\underline{\sigma}$. In addition, our constructions are instance-wise in the sense that $t(\mathsf{F}_{\underline{\rho}(\underline{\sigma})})$ contains only a single instance of $\mathsf{F}_{\underline{\rho}(\underline{\sigma})}$.

The remainder of this section is essentially a survey of the extensions of system T that form the subject of this paper, along with some essential preliminary results. In order to provide some context into the proof theoretic significance of these forms of recursion we also mention some of their applications as bullet points •, although none of these facts are directly relevant or necessary for this work.

2.1 Implicit forms of bar recursion

We begin by introducing the key variant of bar recursion that will feature in our definability results, the so-called *implicit product of selection functions*, or IPS, introduced by Escardó and Oliva in [13]. This is our chosen representative of a class of equivalent bar recursive extensions of system T which includes the more widely known modified bar recursion, but *not* Spector's bar recursion, as we discuss in Section 2.2. One of the characterising features of this class is that the recursion takes place over a *non-computable* tree whose leafs correspond to points of continuity of the parameters.

The defining equation of IPS is given by

$$\mathsf{IPS}^{\varepsilon,q}_{\rho,\tau}(s^{\rho^*}) \overset{\rho^{\mathbb{N}}}{:=} s \mathbin{@} \mathsf{IPS}^{\varepsilon,q}_{\rho,\tau}(s * \varepsilon_s(\lambda x \,.\, q(\mathsf{IPS}^{\varepsilon,q}_{\rho,\tau}(s * x)))) \tag{1}$$

where $\varepsilon \colon \rho^* \to ((\rho \to \tau) \to \rho)$ and $q \colon \rho^{\mathbb{N}} \to \tau$. The type ρ ranges over all finite types, but the type τ is restricted to the discrete types, otherwise the defining axiom becomes inconsistent with $\mathsf{E\text{-}HA}^\omega$, as

we explain below. Because the parameters ε and q don't change during recursive calls we put them in the superscript for clarity, and we omit both types and fixed parameters when there is no danger of ambiguity.

By using the abbreviation $a_s := \varepsilon_s(\lambda x\,.\,q(\mathsf{IPS}^{\varepsilon,q}_{\rho,\tau}(s*x)))$, we can write the defining axiom for IPS a little more concisely as

$$\mathsf{IPS}^{\varepsilon,q}_{\rho,\tau}(s^{\rho^*}) \overset{\rho^{\mathbb{N}}}{:=} s \mathbin{@} \mathsf{IPS}^{\varepsilon,q}_{\rho,\tau}(s * a_s).$$

We use this abbreviation throughout the rest of the paper, beginning with the proofs of the following straightforward but absolutely crucial properties of IPS, which are adapted from analogous properties of Spector's bar recursion highlighted in [29]. We explore the intuition behind these results afterwards.

Lemma 2.3. *Let $\alpha := \mathsf{IPS}^{\varepsilon,q}(s)$. Then $\alpha = \mathsf{IPS}^{\varepsilon,q}([\alpha](n))$ for all $n \geq |s|$.*

Proof. Induction on n. For $n = |s|$ the result is obvious since $[\alpha](|s|) = s$. For the induction step we see that

$$\alpha \overset{I.H.}{=} \mathsf{IPS}([\alpha](n)) = [\alpha](n) \mathbin{@} \mathsf{IPS}([\alpha](n) * a_{[\alpha](n)}) = \mathsf{IPS}([\alpha](n) * a_{[\alpha](n)}),$$

the last equality holding since $[\alpha](n)$ is a prefix of $\mathsf{IPS}([\alpha](n) * a_{[\alpha](n)})$. But then

$$\alpha(n) = \mathsf{IPS}([\alpha](n) * a_{[\alpha](n)})(n) = a_{[\alpha](n)}$$

and so $\alpha = \mathsf{IPS}([\alpha](n+1))$. □

Theorem 2.4. *Define $\alpha\colon \rho^{\mathbb{N}}$ and $p_s\colon \rho \to \tau$ by*

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

Then for all n we have

$$\begin{aligned} \alpha(n) &= \varepsilon_{[\alpha](n)}(p_{[\alpha](n)}) \\ p_{[\alpha](n)}(\alpha(n)) &= q(\alpha). \end{aligned} \tag{2}$$

Proof. For the first equality we have

$$\begin{aligned} \alpha(n) &= \mathsf{IPS}([\alpha](n))(n) \\ &= ([\alpha](n) \mathbin{@} \mathsf{IPS}([\alpha](n) * a_{[\alpha](n)}))(n) \\ &= \varepsilon_{[\alpha](n)}(\lambda x\,.\,q(\mathsf{IPS}([\alpha](n) * x))) \\ &= \varepsilon_{[\alpha](n)}(p_{[\alpha](n)}), \end{aligned}$$

where the first step follows directly from Lemma 2.3. For the second we have

$$p_{[\alpha](n)}(\alpha(n)) = q(\mathsf{IPS}([\alpha](n+1))) \quad = q(\alpha).$$

□

We now pause for a moment to give some insight into the recursor IPS, which can be described quite intuitively using concepts from the world of sequential games. This link between bar recursion and game theory is due to Escardó and Oliva, and is explored in much more detail in e.g. [13, 14], so the reader is invited to consult these works in conjunction with the rapid summary we give here.

Let us first imagine that the type ρ represents a type of 'moves', and τ a type of 'outcomes'. We view infinite sequences $\alpha\colon \rho^{\mathbb{N}}$ as plays in an infinite sequential game, and the parameter q of IPS as

assigning an outcome to each play (because the outcome type τ is always discrete, a continuous q only ever depends on a finite initial segment of each play, ensuring that the game is well-founded).

A strategy $next\colon \rho^* \to \rho$ in a sequential game is a procedure that chooses a next move from a finite initial play $s\colon \rho^*$, and the corresponding strategic extension $\beta_s\colon \rho^{\mathbb{N}}$ of s is the recursive application of $next$ i.e. $\alpha(n) = s_n$ for $n < |s|$ and $\alpha(n) = next([\alpha](n))$ otherwise. The parameter ε of IPS can be seen to represent a *selection function*, that for each finite sequence of moves $s\colon \rho^*$, dictates an optimal next move a_s. A strategy $next$ is optimal if it satisfies

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

where β_{s*x} is the strategic continuation of $s * x$. The strategic continuation of an optimal strategy is an optimal continuation.

Together, ε and q can be viewed as defining a sequential game. In this language, it can be shown as a direct consequence of Lemma 2.3 and the defining equation (1) that $\mathsf{IPS}^{\varepsilon,q}$ is a backtracking recursor with the property that $\mathsf{IPS}(s)$ is the optimal continuation of the finite initial play s in this game. Theorem 2.4 simply makes it explicit that $\alpha = \mathsf{IPS}(\langle\rangle)$ is optimal at every point in the sequence.

In [13, 14], IPS is constructed in a slightly more elaborate way as an infinite iteration of a binary product operation on selection functions - from which its name derives. This makes the game-semantics of the recursor more perspicuous. Here, on the other hand, our chief concern is syntactic simplicity, so we avoid this construction and define IPS directly without resorting to any additional machinery, although our variant is completely equivalent (see [25, Appendix A] for details). While the rather technical instances of IPS considered in the paper do not correspond to particularly natural games, a game-theoretic reading of IPS can be extremely helpful for understanding the basic behaviour of this form of recursion.

On a more abstract level, the equations (2) characterise IPS as a functional that constructs a kind of 'sequential equilibrium' between the selection functions ε_s. Optimal plays in sequential games form one example of this kind of equilibrium, but it also appears in other contexts, most importantly in giving a computational content to the axiom of choice. The basic idea here is that the ε_s serve as 'local' realizers for the premise of the axiom and the bar recursion is used to combine them into a 'global' a realizer α for the conclusion. Again, this is all discussed in considerable detail in elsewhere, and here we conclude our brief discussion by simply stating together the following striking pair of applications of IPS.

- IPS solves the modified realizability interpretation of the negative translation of both countable and countable dependent choice, and can therefore be used to extract computational content from proofs in classical analysis [12, 16].
- IPS computes Nash-equilibria in a general class of implicitly-controlled infinite sequential games, as defined in [14, 15].

Before we move on we quickly show that IPS can also be expressed as a solution to an alternative, but equivalent defining axiom, which is essentially an unwinding of Lemma 2.3. This alternative formulation of IPS is extremely convenient, and in the remainder of the paper we freely use the fact that IPS satisfies both (1) and (3) below.

Lemma 2.5. *The implicit product of selection functions can be alternatively defined using course-of-values recursion as*

$$\mathsf{IPS}^{\varepsilon,q}(s) \overset{\rho^{\mathbb{N}}}{=} s \mathbin{@} \lambda n\,.\, \varepsilon_{t_n}(\lambda x\,.\, q(\mathsf{IPS}^{\varepsilon,q}(t_n * x))) \tag{3}$$

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

Proof. What we mean by this is that any functional satisfying the defining axiom (1) also satisfies (3) and vice-versa. Suppose that IPS is defined as in (1). Then $\mathsf{IPS}(s)(n) = s_n$ for $n < |s|$, and for $n \geq |s|$ we have, by Lemma 2.3,

$$\mathsf{IPS}(s)(n) = (t_n \mathbin{@} \mathsf{IPS}(t_n * a_{t_n}))(n) = a_{t_n} = \varepsilon_{t_n}(\lambda x \,.\, q(\mathsf{IPS}(t_n * x))).$$

Therefore IPS satisfies (3). For the converse, if IPS is defined as in (3), let $t_n := [\mathsf{IPS}(s)](n)$ and $\tilde{t}_n := [\mathsf{IPS}(s * a_s)](n)$ for $a_s := \varepsilon_s(\lambda x \,.\, q(\mathsf{IPS}(s * x)))$. We show by induction that $t_n = \tilde{t}_n$ for all n. Clearly this is true for $n < |s|$, and for $n = |s|$ we have $\mathsf{IPS}(s)(|s|) = a_s = \mathsf{IPS}(s * a_s)(|s|)$ by definition. Finally, for $n \geq |s|$ we have

$$\mathsf{IPS}(s)(n) = \varepsilon_{t_n}(\lambda x \,.\, q(\mathsf{IPS}(t_n * x))) \overset{I.H.}{=} \varepsilon_{\tilde{t}_n}(\lambda x \,.\, q(\mathsf{IPS}(\tilde{t}_n * x))) = \mathsf{IPS}(s * a_s)(n)$$

and therefore $t_{n+1} = \tilde{t}_{n+1}$. Thus $\mathsf{IPS}(s) = \mathsf{IPS}(s * a_s)$, and since s is a prefix of $\mathsf{IPS}(s * a_s)$ we have $\mathsf{IPS}(s) = s \mathbin{@} \mathsf{IPS}(s * a_s)$. □

Finally, we pause again to explain why IPS exists in the type-structure $\mathcal{C}^\omega$ of continuous functionals - by which we mean that for each ρ, τ there exists an element $\Phi_{\rho,\tau}$ of $\mathcal{C}_\sigma$ (where σ is the type of $\mathsf{IPS}_{\rho,\tau}$) such that $\mathsf{IPS}^{\mathsf{ax}}_{\rho,\tau}[\Phi_{\rho,\tau}]$ holds in $\mathcal{C}^\omega$. Recursive extensions of system T can usually be justified in the continuous functionals via a domain theoretic argument (see e.g. [4, 5]), where by Ershov's well-known result [10] one identifies the total continuous functionals $\mathcal{C}^\omega$ as the extensional collapse of the total elements of the partial continuous functionals $\hat{\mathcal{C}}^\omega$. The model $\hat{\mathcal{C}}^\omega$ has the property that *any* continuous functional $\hat{\mathcal{C}}_\sigma \to \hat{\mathcal{C}}_\sigma$ has a fixpoint, and so in particular any functional defined as a fixpoint of a primitive recursive equation has an interpretation in $\hat{\mathcal{C}}^\omega$. To show that it exists in $\mathcal{C}^\omega$ it is sufficient to prove that a solution in $\hat{\mathcal{C}}^\omega$ is total. For IPS this can be done via the following reasoning:

> Suppose for contradiction that there are total arguments ε, q and s such that $\mathsf{IPS}^{\varepsilon,q}(s)$ is not total in $\hat{\mathcal{C}}^\omega$. Then there must be some total x_0 such that $\mathsf{IPS}^{\varepsilon,q}(s * x_0)$ is also not total, and repeating this argument, by dependent choice there exists a total sequence α such that $\mathsf{IPS}(s * [\alpha](n))$ is not total for each n.
>
> But since τ is discrete, by the continuity axiom in $\hat{\mathcal{C}}^\omega$ there is a point N such that $q(s * [\alpha](N) \mathbin{@} \beta) = q(s * \alpha)$ for any sequence β, and therefore by Lemma 2.5
>
> $$\mathsf{IPS}(s * [\alpha](N)) = s * [\alpha](N) \mathbin{@} \lambda n \,.\, \varepsilon_{t_n}(\lambda x \,.\, q(s * \alpha))$$
>
> which is total, and thus contradicts the construction of α.

Note that without the restriction on the type τ being discrete, this argument would no longer be valid. A straightforward counterexample is given by $\mathsf{IPS}_{\mathbb{N},\mathbb{N}\to\mathbb{N}}$, whose defining axiom is no longer even consistent with $\mathsf{E\text{-}HA}^\omega$. This can be seen by setting $q\colon \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ as the identity, and $\varepsilon_s(p^{\mathbb{N}\to\mathbb{N}^\mathbb{N}}) = p(0)(|s|+1)+1$. Then there is no natural number n satisfying $n = \mathsf{IPS}(\langle\rangle)(0)$.

We now move on to discuss some equivalent formulations of implicit bar recursion, the first of which in particular is crucial for later sections.

2.1.1 The simple product of selection functions

The simple implicit product of selection functions ips has defining equation

$$\mathsf{ips}^{\varepsilon,q}_{\rho,\tau}(s) = s \mathbin{@} \mathsf{ips}^{\varepsilon,q}_{\rho,\tau}(s * a_s)$$

where now ε has type $\mathbb{N} \to ((\rho \to \tau) \to \rho)$ and $a_s := \varepsilon_{|s|}(\lambda x \,.\, q(\mathsf{ips}(s * x)))$. This differs from IPS in that now the selection functions $\varepsilon_{|s|}$ only have access to the length of s, not the values of its elements. Note that as with the dependent product we can rephrase ips using course-of-values recursion as

$$\mathsf{ips}^{\varepsilon,q}(s) = s \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, q(\mathsf{ips}(t_n * x)) \tag{4}$$

where $t_n := [\mathsf{ips}(s)](n)$. The simple product has a significance of its own in proof theory as it is sufficient to give a computational interpretation to countable (as opposed to countable dependent) choice:

- ips solves the modified realizability interpretation of the negative translation of countable choice [12].

In [17] it is proven that ips is actually primitive recursively equivalent to the full product IPS, although one must raise the type of ips to obtain the equivalence. This fact means that in order to define IPS it is sufficient to define the simple product ips. We take advantage of this in Section 4 when defining IPS from BBC, and so given its importance in this paper we outline the equivalence proof below.

The proof also forms a convenient warm up for later proofs, which follow the same general pattern of a main construction followed by a routine and somewhat tedious verification involving $\mathsf{Cont} + \mathsf{QF\text{-}BI}$, seemingly necessary to confirm that certain inessential alterations of the arguments of the bar recursion have no effect on the outcome.

Theorem 2.6 (Escardó-Oliva [17])**.** $\mathsf{ips}_{\rho^* \to \rho, \tau} \geq_{\mathsf{T}} \mathsf{IPS}_{\rho,\tau}$ *over* $\mathsf{Cont} + \mathsf{QF\text{-}BI}$.

Proof. We rephrase the proof given in [17] so that it applies to our slightly altered definition of IPS and ips. For parameters $\varepsilon\colon \rho^* \to ((\rho \to \tau) \to \rho)$ and $q\colon \rho^{\mathbb{N}} \to \tau$ for $\mathsf{IPS}_{\rho,\tau}$, define

$$\begin{aligned} \tilde{\varepsilon}_n(p^{(\rho^* \to \rho) \to \tau}) &\overset{\rho^* \to \rho}{=} \lambda s \,.\, \varepsilon_s(\lambda x \,.\, p(\lambda u.x)) \\ \tilde{q}(\alpha^{(\rho^* \to \rho)^{\mathbb{N}}}) &:\overset{\tau}{=} q(\{\alpha\}) \end{aligned}$$

where $\{\alpha\}\colon \rho^{\mathbb{N}}$ is defined recursively as $\{\alpha\}(n) := \alpha(n)([\{\alpha\}](n))$ (and similarly for finite sequences). Now set

$$\Phi^{\varepsilon,q}_{\rho,\tau}(s) \overset{\rho^{\mathbb{N}}}{:=} \{\mathsf{ips}^{\tilde{\varepsilon},\tilde{q}}_{\rho^* \to \rho,\tau}(\bar{s})\}$$

where $\bar{s}_n := \lambda u.s_n$. We claim that $\mathsf{IPS}^{\mathsf{ax}}_{\rho,\tau}[\Phi_{\rho,\tau}]$. Setting $t_n = [\Phi(s)](n) = [\{\mathsf{ips}(\bar{s})\}](n)$ and $T_n = [\mathsf{ips}(\bar{s})](n)$, for $n < |s|$ we have

$$\Phi(s)(n) = \mathsf{ips}(\bar{s})(n)(t_n) = \bar{s}_n(t_n) = s_n,$$

while for $n \geq |s|$ we have (using (5))

$$\begin{aligned} \Phi(s)(n) &= \mathsf{ips}(\bar{s})(n)(t_n) \\ &= \tilde{\varepsilon}_n(\lambda f^{\rho^* \to \rho} \,.\, q(\{\mathsf{ips}(T_n * f)\}))(t_n) \\ &= \varepsilon_{t_n}(\lambda x^{\rho} \,.\, q(\{\mathsf{ips}(T_n * \lambda u.x)\})) \\ &\overset{(+)}{=} \varepsilon_{t_n}(\lambda x \,.\, q(\{\mathsf{ips}(\overline{\{T_n * \lambda u.x\}})\})) \\ &\overset{(i)}{=} \varepsilon_{t_n}(\lambda x \,.\, q(\{\mathsf{ips}(\overline{t_n * x})\})) \\ &= \varepsilon_{t_n}(\lambda x \,.\, q(\Phi(t_n * x))). \end{aligned}$$

Here (i) follows from the fact that $\{T_n * \lambda u.x\} = t_n * x$ (a simple induction argument), while $(+)$ is derived using QF-BI on the formula

$$P(R) :\equiv q(\{\mathsf{ips}(T * R)\}) = q(\{\mathsf{ips}(\overline{\{T\}} * R)\})$$

for $T := T_n * \lambda u.x$. For arbitrary $\beta\colon (\rho^* \to \rho)^{\mathbb{N}}$, let N be a point of continuity of $q_{\{T\}}$ on $\mathsf{tail}_{|T|}\{T * \beta\}$ (where $\mathsf{tail}_n(\alpha) := \lambda k.\alpha(n+k)$). Then

$$\begin{aligned} q(\{\mathsf{ips}(T * [\beta](N))\}) &= q(\{T * [\beta](N) \mathbin{@} \ldots\}) \\ &= q([\{T * \beta\}](|T| + N) \mathbin{@} \ldots) \\ &= q(\{T\} * [\mathsf{tail}_{|T|}\{T * \beta\}](N) \mathbin{@} \ldots) \end{aligned}$$

and similarly

$$\begin{aligned} q(\{\mathsf{ips}(\overline{\{T\}} * [\beta](N))\}) &= q(\{\overline{\{T\}} * [\beta](N) \mathbin{@} \ldots\}) \\ &= q([\{T * \beta\}](|T| + N) \mathbin{@} \ldots) \\ &= q(\{T\} * [\mathsf{tail}_{|T|}\{T * \beta\}](N) \mathbin{@} \ldots) \end{aligned}$$

and therefore $P([\beta](N))$ holds by Cont. For the induction step, assuming as bar induction hypothesis $\forall f P(R * f)$, we have

$$\begin{aligned} q(\{\mathsf{ips}(T * R)\}) &= q(\{\mathsf{ips}(T * R * a_{T*R})\}) \\ &\overset{(ii)}{=} q(\{\mathsf{ips}(T * R * a_{\overline{\{T\}}*R})\}) \\ &\overset{B.I.H.}{=} q(\{\mathsf{ips}(\overline{\{T\}} * R * a_{\overline{\{T\}}*R})\}) \\ &= q(\{\mathsf{ips}(\overline{\{T\}} * R)\}) \end{aligned}$$

where for (ii) we have

$$\begin{aligned} a_{T*R} &= \tilde{\varepsilon}_{|T|+|R|}(\lambda f\,.\, q(\{\mathsf{ips}(T * R * f)\})) \\ &\overset{B.I.H.}{=} \tilde{\varepsilon}_{|T|+|R|}(\lambda f\,.\, q(\{\mathsf{ips}(\overline{\{T\}} * R * f)\})) \\ &= a_{\overline{\{T\}}*R}. \end{aligned}$$

We have established $\forall\beta\exists N P([\beta](N))$ and also $\forall R(\forall f P(R*f) \to P(R))$, so by bar induction (trivially relativised) we obtain $P(\langle\rangle)$ and hence $(+)$. □

2.1.2 Modified bar recursion

The implicit product of selection functions is predated by the more widely known modified bar recursion [5, 6], which is in turn based on the realizer for dependent choice given in [1] (not to be confused with the BBC functional of the same paper). Modified bar recursion has the defining equation

$$\mathsf{MBR}^{\psi,q}_{\rho,\tau}(s^{\rho^*}) :\overset{\tau}{=} q(s \mathbin{@} \psi_s(\lambda x^{\rho}\,.\,\mathsf{MBR}(s * x)))$$

for $\psi\colon \rho^* \to ((\rho \to \tau) \to \rho^{\mathbb{N}})$ and $q\colon \rho^{\mathbb{N}} \to \tau$, where as with IPS the type τ is required to be discrete. In [1, 5] MBR is shown to interpret the axiom of dependent choice, while its relationship to other forms of bar recursion is analysed in [6]. Strong normalization for MBR is established in [4], while concrete applications for the extraction of programs in mathematical analysis are given in e.g. [27, 28]. It is not too difficult to show that IPS is interdefinable with MBR.

Theorem 2.7. $\mathsf{IPS} =_{\mathsf{T}} \mathsf{MBR}$ *instance-wise over* $\mathsf{Cont} + \mathsf{QF\text{-}BI}$.

Proof. This is proved in [17, §5]. □

2.2 Spector's bar recursion

In the previous section we introduced the implicit product of selection functions IPS and showed that in terms of primitive recursive equivalence it has the same computational strength as its simple version ips and modified bar recursion MBR. In order to put this paper's main results in context and to present a more complete picture of the classification of bar recursors, in this section we briefly discuss various *explicit* forms of bar recursion that turn out to be weaker than the implicit forms.

Perhaps the most well-known of this kind of bar recursion is Spector's bar recursion [29], which in its most general form has defining equation

$$\mathsf{GBR}^{\phi,r,\varphi}_{\rho,\tau}(s^{\rho^*}) \stackrel{\tau}{=} \begin{cases} r(s) & \text{if } \varphi(\widehat{s}) < |s| \\ \phi_s(\lambda x^\rho \,.\, \mathsf{GBR}^{\phi,r,\varphi}(s*x)) & \text{otherwise.} \end{cases}$$

where $\phi\colon \rho^* \to ((\rho \to \tau) \to \tau)$, $r\colon \rho^* \to \tau$ and $\varphi\colon \rho^{\mathbb{N}} \to \mathbb{N}$. Spector's bar recursion exists in continuous models due the fact that infinite sequences α eventually reach a point n such that $\varphi(\overline{\alpha,n}) < n$ (this is a direct consequence of Cont), although it also exists in the type structure of strongly majorizable functionals $\mathcal{M}^\omega$ that contains non-continuous functionals [7]. Because the underlying tree is given explicitly in this way, the outcome type τ can now be arbitrary.

Spector also identified a special instance SBR_ρ of bar recursion (definable from $\mathsf{GBR}_{\rho,\rho^{\mathbb{N}}}$) that has defining equation

$$\mathsf{SBR}^{\phi,\varphi}_{\rho}(s^{\rho^*}) \stackrel{\rho^{\mathbb{N}}}{=} s \mathbin{@} \begin{cases} \mathbf{0} & \text{if } \varphi(\widehat{s}) < |s| \\ \mathsf{SBR}^{\phi,\varphi}(s*a_s) & \text{otherwise} \end{cases}$$

for $a_s := \phi_s(\lambda x \,.\, \mathsf{SBR}(s*x))$. More recently Escardó and Oliva cast these variants in terms of their products of selection functions, showing that GBR is essentially a product of *quantifiers* EPQ (see [17, 14]) while SBR is directly equivalent to an *explicit* product of selection functions given by

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

where $a_s := \varepsilon_s(\lambda x \,.\, q(\mathsf{EPS}(s*x)))$. This latter form of recursion is clearly analogous to IPS - the only difference being the addition of an explicit stopping condition - and as with IPS the special variants of Spector's bar recursion have a profound significance in both proof theory and the theory of generalised sequential games.

- Both SBR and EPS solve the Dialectica interpretation of the negative translation of countable and countable dependent choice (originally due to Spector [29], adapted for EPS in [12]).
- EPS computes Nash-equilibria in a general class of explicitly-controlled infinite sequential games, as defined in [14, 15].

Remark 2.8. As with IPS, one can define a simple (and equivalent) form eps of EPS, which is sufficient to solve the Dialectica interpretation of the negative translation of countable choice.

Theorem 2.9. *All of the above variants of Spector's bar recursion are primitive recursively equivalent over* $\mathsf{QF\text{-}BI} + \mathsf{Spec}$*, where*

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

In particular, since $\mathcal{C}^\omega, \mathcal{M}^\omega \models \mathsf{Spec}$ *these equivalences hold in both* $\mathcal{C}^\omega$ *and* $\mathcal{M}^\omega$*.*

Proof. The equivalences $\mathsf{EPS} =_{\mathsf{T}} \mathsf{SBR}$ and $\mathsf{EPQ} =_{\mathsf{T}} \mathsf{GBR}$ are trivial, while $\mathsf{eps} =_{\mathsf{T}} \mathsf{EPS}$ is due to [17] and $\mathsf{SBR} =_{\mathsf{T}} \mathsf{GBR}$ is due to [24]. □

A number of further formulations of GBR that appeared in the decades following Spector's paper were shown to be equivalent in [8]. Together with Theorem 2.9 the result is that we now know that most variants of Spector's bar recursion are equivalent.

2.2.1 Implicit bar recursion is strictly stronger than explicit bar recursion

It was first proved by Berger and Oliva [6] that while modified bar recursion defines Spector's bar recursion, the converse is not true. The reasons for this are rather technical, but we do our best to outline them in brief below. The reader is encouraged to consult the original papers for details.

It is a well known fact that there are elements of the model $\mathcal{C}^\omega$ which have recursive associates but are not S1-S9 computable in $\mathcal{C}^\omega$. The canonical example of this is the FAN functional which locates points of uniform continuity for functionals on boolean sequences:

$$\mathsf{FAN}[\Phi] \ : \ \forall q^{\mathbb{B}^{\mathbb{N}}\to\mathbb{N}}, \alpha, \beta([\alpha](\Phi(q)) = [\beta](\Phi(q)) \to q(\alpha) = q(\beta)).$$

A more extreme example is the so-called Γ-functional of Gandy and Hyland [18] defined by

$$\Gamma^q(s^{\mathbb{N}^*}) \stackrel{\mathbb{N}}{=} q(s * 0 * \lambda n \,.\, \Gamma(s * (n+1))),$$

which despite having a simple recursive defining equation is not S1-S9 computable even in the FAN functional. Berger and Oliva [6] prove that MBR of lowest type is equivalent to the Γ-functional, and therefore MBR cannot be S1-S9 computable either (in the same paper they also give a direct proof of this fact by showing that MBR S1-S9 defines the FAN functional).

On the other hand, GBR is S1-S9 definable in $\mathcal{C}^\omega$ by the recursion theorem, so by the fact that S1-S9 computable functionals are closed under primitive recursion we obtain:

Theorem 2.10 (Berger-Oliva [6])**.** MBR *is not primitive recursively definable from* GBR *over any theory* Δ *validated by* $\mathcal{C}^\omega$*.*

As an immediate corollary it holds more generally that no extension of T primitive recursively equivalent to MBR in $\mathcal{C}^\omega$ is definable from an extension of T primitive recursively equivalent to GBR in $\mathcal{C}^\omega$, so in particular IPS is not definable from GBR.

It may seem strange that IPS and MBR are defined recursively (which is permitted in S1-S9) but nevertheless not computable like GBR. The subtle reason for this lies in the S8 rule for functional application, which requires the input to be a total object before the outcome is total. One can show that a fixpoint of the defining equation for GBR is indeed a computable functional by bar induction, since eventually we reach a point such that $\mathsf{GBR}(s) = r(s)$ and $r(s)$ is total. However, a fixpoint for MBR cannot be shown to be computable in the same way: while at points of continuity we may indeed have $\mathsf{MBR}(s) = q(s \mathbin{@} \psi_s(\lambda x \,.\, \mathsf{MBR}(s * x))) = q(\widehat{s})$, this reasoning does not work in the S1-S9 rules because S8 requires all of $s \mathbin{@} \psi_s(\lambda x \,.\, \mathsf{MBR}(s * x))$ to be defined before it returns a value for $\mathsf{MBR}(s)$. Thus S1-S9 computability in $\mathcal{C}^\omega$ separates the explicit from the implicit forms of bar recursion.

2.2.2 Kohlenbach's bar recursion

A somewhat different instance of explicit bar recursion is given by Kohlenbach in [20] and also discussed in detail in [5, 6]. This has the defining equation

$$\mathsf{KBR}^{\phi,r,\varphi}_{\rho,\tau}(s^{\rho^*}) \stackrel{\tau}{=} \begin{cases} r(s) & \text{if } \varphi(\widehat{s}) = \varphi(\check{s}) \\ \phi_s(\lambda x \,.\, \mathsf{KBR}^{\phi,r,\varphi}(s * x)) & \text{otherwise.} \end{cases}$$

where $\check{s} = s \mathbin{@} \mathbf{1}_{\rho^{\mathbb{N}}}$. As with GBR, the presence of an explicit stopping condition means that KBR it is S1-S9 computable and therefore does not define MBR. However, the other direction also fails to hold as $\mathcal{M}^\omega$ is known to be a model of MBR [5] but not of KBR (for the same reason, KBR is not definable from GBR either). One can take two important morals from KBR: Firstly it is a straightforward example of a mode of recursion that is disjoint from both implicit bar recursion and Spector's bar recursion, highlighting the fact that these classes far from encompass all recursive extensions of system T, and secondly that even a small adjustment to the defining equations of a form of recursion (in this case a simple alteration of the stopping condition) can lead to a very different extensions of system T. In many cases it is not at all obvious from the defining equations whether or not given extensions of T are equivalent.

2.3 The BBC functional and open recursion

We now define the second main subject of our paper, namely a family of extensions of system T based not on bar recursion, but open recursion. By open recursion we mean recursion over the lexicographic ordering on infinite sequences. However, before we define and discuss this form of recursion properly, we introduce the instance that is perhaps most well-known in proof theory - the Berardi-Bezem-Coquand realizer of countable choice [1], which we have abbreviated the BBC functional.

In one sentence, the BBC functional is an alternative to bar recursion, that instead of making recursive calls on extensions of finite sequences, makes them on *updates* of partial sequences. Let us make this more precise: Given a partial sequence $u\colon \bar{\rho}^{\mathbb{N}}$ (cf. Section 1.1 for the type $\bar{\rho}$) and an element $x\colon \rho$, the partial sequence u_n^x is defined by

$$u_n^x(m) :\overset{\bar{\rho}}{=} \begin{cases} x & \text{if } m = n \\ u(m) & \text{otherwise.} \end{cases}$$

We call u_n^x an *update* of u whenever $n \notin \mathrm{dom}(u)$. The BBC functional has the defining equation

$$\mathsf{BBC}_{\rho,\tau}^{\varepsilon,q}(u^{\bar{\rho}^{\mathbb{N}}}) \overset{\rho^{\mathbb{N}}}{=} u \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, q(\mathsf{BBC}^{\varepsilon,q}(u_n^x)))$$

where $\varepsilon\colon \mathbb{N} \to ((\rho \to \tau) \to \rho)$ and $q\colon \rho^{\mathbb{N}} \to \tau$, and as with implicit bar recursion the type τ is required to be discrete.

Remark 2.11. The BBC functional is often defined slightly differently in the literature. In particular in [1] it takes as input only partial sequences with finite domain. We follow [2] in generalising it to allow partial functions with arbitrary domain as this appears to be a slightly more elegant formulation that can be directly related to open recursion, although we point out that our main proof that BBC defines ips in section 4 only uses an instance of BBC with finite domain.

Despite having a relatively simple defining equation, the BBC functional can be rather elusive to understand. Let us recall the defining equation of the simple variant of implicit bar recursion, which is

$$\mathsf{ips}^{\varepsilon,q}(s^{\rho^*}) = s \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, q(\mathsf{ips}(t_n * x))) \tag{5}$$

where $t_n := [\mathsf{ips}(s)](n)$. This is the form of implicit bar recursion most obviously related to BBC, and by putting them side by side we are able to see more clearly the essential differences between bar recursion and the BBC functional.

To begin with, both forms of recursion are related in that they complete a partially defined input by using the selection functions ε_n, and make recursive calls 'backwards' by adding one more piece of information to this input. However, while for bar recursion this input is always a finite sequence s, and

recursive calls are made on extensions $s * x$ of this sequence, for the BBC functional the input can be an *arbitrary* partial function u, and recursive calls are made over updates u_n^x of this partial function. This leads to a form of recursion that behaves very differently to bar recursion.

One striking feature of this difference is the 'forgetful' nature of BBC. The sequence $\alpha = \mathsf{ips}(\langle\rangle)$ is computed sequentially: from (5) we have

$$\alpha(n) = \varepsilon_n(\lambda x \,.\, q(\mathsf{ips}(\langle \alpha(0), \ldots, \alpha(n-1), x\rangle))),$$

in other words, once $\langle \alpha(0), \ldots, \alpha(n-1)\rangle$ are computed they are fixed and all later entries are computed relative to this initial segment. On the other hand, for $\beta = \mathsf{BBC}(\emptyset)$ (where $\emptyset$ is the partial function undefined everywhere) we have

$$\beta(n) = \varepsilon_n(\lambda x \,.\, q(\mathsf{BBC}(\emptyset_n^x))),$$

in other words, the values of $\langle \beta(0), \ldots, \beta(n-1)\rangle$ are ignored when computing $\beta(n)$, which is done via a separate recursion tree.

One can see this more concretely by considering simple binary versions $\mathsf{ips}_{\mathrm{bin}}$ and $\mathsf{BBC}_{\mathrm{bin}}$, in which the value of the parameter q depends only on the first two elements of its input. Let $\langle a_0, a_1\rangle := \mathsf{ips}_{\mathrm{bin}}(\langle\rangle)$ and $\langle b_0, b_1\rangle := \mathsf{BBC}_{\mathrm{bin}}(\langle\rangle)$. By unwinding the definitions we see that

$$a_0 = \varepsilon_0(\lambda x \,.\, q(x, \varepsilon_1(\lambda y \,.\, q(x, y)))) = b_0,$$

but while $a_1 = \varepsilon_1(\lambda x \,.\, q(a_0, x))$, the value of b_1 does not depend on b_0 but is calculated symmetrically as

$$b_1 = \varepsilon_1(\lambda x \,.\, q(\varepsilon_0(\lambda y \,.\, q(y, x)), x)).$$

The BBC functional was originally conceived as an alternative way of giving a computational interpretation to classical analysis, where its 'demand-driven' manner of ignoring information that it doesn't need was seen as a clear advantage over bar recursion, which is always forced to compute every value up to the point in question:

- BBC solves the modified realizability interpretation of the negative translation of countable choice (due to [1] but see also [2]).

This result also highlighted the arbitrary nature of bar recursion, in that its widespread use in giving a computational interpretation to classical analysis was largely down to convention, while interesting and potentially better alternatives for the purpose of program extraction could be developed. Nevertheless, over the years it is bar recursive extensions of system T that have received the majority of attention in proof theory and computability theory, while in particular the BBC functional, arguably no less interesting from a mathematical perspective than implicit bar recursion, has remained poorly understood. Only later in [2] was it shown to even exist in a standard domain-theoretic model, and so far it has eluded any convincing game semantics along the lines of Escardó and Oliva [14]. However, an important step towards establishing at the very least a secure theoretic framework for the BBC functional was made by Berger [3], who demonstrated that BBC is just a simple instance of open recursion: recursion over the lexicographic ordering on sequences.

2.3.1 Open recursion and update recursion

Suppose that $<\colon \rho \times \rho \to \mathbb{B}$ is a well-founded, decidable binary relation. The lexicographic ordering $<_{\mathrm{lex}}$ over $<$ is the binary relation on infinite sequences defined by

$$\alpha <_{\mathrm{lex}} \beta :\equiv \exists n([\alpha](n) = [\beta](n) \wedge \alpha(n) < \beta(n)).$$

A general schema of open recursion in all finite types was formulated in [3] as

$$\mathsf{OR}^F_{(\rho,<),\tau}(\alpha^{\rho^{\mathbb{N}}}) \stackrel{\tau}{=} F_\alpha(\lambda n, y, \beta\,.\,\mathsf{OR}^F([\alpha](n) * y \mathbin{@} \beta) \text{ if } y < \alpha(n)),$$

where $F\colon \rho^{\mathbb{N}} \times (\mathbb{N} \times \rho \times \rho^{\mathbb{N}} \to \tau) \to \tau$ and τ is restricted to being discrete. At first sight, it is not clear that open recursion should exist even in continuous models, since the lexicographic ordering is not well-founded apart from in trivial cases. For example, for the well-founded relation $0 < 1$ on $\mathbb{B}$, we have the following infinite chain over $<_{\text{lex}}$:

$$10000\ldots >_{\text{lex}} 01000\ldots >_{\text{lex}} 00100\ldots >_{\text{lex}} \ldots$$

and so in particular, induction over $<_{\text{lex}}$ is not generally valid. However, induction over $<_{\text{lex}}$ *is* meaningful - and classically derivable from the axiom of dependent choice - provided that we restrict ourselves to a class of *open* formulas. Induction over $<_{\text{lex}}$ for open predicates was first considered by Raoult [26], and later studied by Coquand [9] and Berger [3], among others. In the language of all finite types, it is given by the schema

$$\mathsf{OI}_{(\rho,<)}\ :\ \forall\alpha(\forall\beta{<_{\text{lex}}}\alpha O(\beta) \to O(\alpha)) \to \forall\alpha O(\alpha).$$

In informal terms, an open formula is one whose negation is a piecewise property on finite initial segments of α i.e. $\neg O(\alpha) \leftrightarrow \forall n B([\alpha](n))$ for some predicate B. In this case, the contrapositive of open induction follows directly from the minimal-bad-sequence of Nash-William [23], and indeed open induction can be seen precisely as an inductive analogue of the minimal-bad-sequence construction (see [3] for details). For this reason, open induction has a deep significance in mathematics, and in particular it is a form of induction that allows us to directly prove both the famous Kruskal tree theorem and Higman's lemma [23]. Open recursion is a computational analogue of open induction, and in [3] it is shown to give a direct realizability interpretation to open induction.

We refrain from going into any further detail here on the proof theoretic background of open recursion. Our aim is simply to convince the reader that it is an interesting and important extension of system T, firstly because it is related to a key induction principle, and secondly, of course, because it is in some sense a generalised form of the BBC functional.

Returning now to the existence of open recursion, it is reasonable to think that if open induction is classically valid for a restricted class of formulas, then open recursion is well-founded in some analogously restricted setting. This is precisely why we insist on the outcome type τ being discrete. In this case, the totality of $\mathsf{OR}(\alpha)$ becomes an open predicate, and totality of OR can be proven by open induction, via the following informal argument (due to [3]):

> Given a total argument F, the statement
>
> $$O(\alpha) :\equiv \alpha \text{ is total} \to \mathsf{OR}^F(\alpha) \text{ is total}$$
>
> is an open predicate, since by the continuity axiom and the fact that τ is discrete, totality of $\mathsf{OR}^F(\alpha)$ depends only on an initial segment of α. But for α total, $\forall\beta{<_{\text{lex}}}\alpha\ O(\beta)$ clearly implies totality of $\mathsf{OR}^F(\alpha)$, therefore totality of OR^F follows from open induction.

We conclude this section by finally explaining how open recursion is linked to the BBC functional. In [3], Berger identifies an important special case of open induction over partial sequences called update induction, given by

$$\mathsf{UI}_\rho\ :\ \forall u^{\bar{\rho}^{\mathbb{N}}}(\forall v <_{\text{up}} u\ O(v) \to O(u)) \to \forall u O(u)$$

where O ranges over open predicates. Here $v <_{\mathrm{up}} u$ when v is an update of u i.e. is of the form u_n^x for $n \notin \mathrm{dom}(u)$. Update induction follows from an instance of open induction of type $\bar{\rho}$ over the trivially well-founded relation $\bot < x$ for $x \in \rho$. Analogously, one can define the update recursor UR as

$$\mathsf{UR}^G_{\rho,\tau}(u^{\bar{\rho}^{\mathbb{N}}}) \stackrel{\tau}{=} G_u(\lambda n, x\,.\,\mathsf{UR}(u_n^x) \text{ if } n \notin \mathrm{dom}(u))$$

where $G\colon \bar{\rho}^{\mathbb{N}} \times (\mathbb{N} \times \rho \to \tau) \to \tau$ and τ is discrete. The following proof theoretic results are all due to [3]:

- Open recursion solves the modified realizability interpretation of open induction;
- Update recursion solves the modified realizability interpretation of update induction;
- The negative translation of countable choice is intuitionistically derivable from update induction, and the corresponding update-recursive realizer of countable choice is the BBC-functional.

The last of these results is striking as it reveals BBC to be a special case of update recursion that naturally arises from the proof that UI implies countable choice. On the level of definability, the relationship between all three forms of recursion can be given as follows.

Theorem 2.12. $\mathsf{OR}_{(\bar{\rho},<),\tau} \geq_{\mathsf{T}} \mathsf{UR}_{\rho,\tau} \geq_{\mathsf{T}} \mathsf{BBC}_{\rho,\tau}$ *instance-wise, where the relation* $<$ *on* $\bar{\rho}$ *is given by* $\bot < x$ *for all* $x \in \rho$.

Proof. For the first inequality we define $\Phi^G_{\rho,\tau}(u) := \mathsf{OR}^{\tilde{F}}_{\bar{\rho},\tau}(u)$ where

$$\tilde{F}_u(P^{\mathbb{N}\times\bar{\rho}\times\bar{\rho}^{\mathbb{N}}}) := G_u(\lambda n, x^\rho\,.\,Pnxu)$$

and it is straightforward to show that $\mathsf{UR}^{\mathsf{ax}}_{\rho,\tau}[\Phi_{\rho,\tau}]$. For the second inequality we set

$$\Psi^{\varepsilon,q}_{\rho,\tau}(u) := u \mathbin{@} \lambda n\,.\,\varepsilon_n(\lambda x\,.\,\mathsf{UR}^{\tilde{G}}_{\rho,\tau}(u_n^x))$$

where

$$\tilde{G}_u(P^{\mathbb{N}\times\rho\to\tau}) := q(u \mathbin{@} \lambda n\,.\,\varepsilon_n(\lambda x\,.\,Pnx))$$

and again it is straightforward to prove that $\mathsf{BBC}^{\mathsf{ax}}_{\rho,\tau}[\Psi_{\rho,\tau}]$. □

2.4 Summary

In Sections 2.1 and 2.2 we gave short survey of some of the work done towards understanding and classifying the many variants of bar recursion, while in Section 2.3 we outlined similar work in the context of open recursive functionals. However, none of this research addresses the relationship between bar recursion and open recursion. That is the purpose of this paper, and in the remaining sections we show that open recursion, update recursion and the BBC functional are in fact all equivalent to modified bar recursion.

3 Defining open recursion from IPS

We first show that open recursion is definable from implicit bar recursion. Before we do so we outline our construction on an informal level, for the simple case of the Cantor space, although the reader is encouraged to skim through this description at first and consult it in conjunction with the main proof that follows.

Suppose that we want to build a term $\Phi = t(\mathsf{IPS})$ satisfying the defining equation for open recursion on $\mathbb{B}^{\mathbb{N}}$ (lexicographically ordered based on the relation $0 < 1$), which is

$$\Phi^F(\alpha) = F_\alpha(\lambda n, y, \beta\,.\,\Phi([\alpha](n) * y \mathbin{@} \beta) \text{ if } y = 0 \wedge \alpha(n) = 1).$$

Our main idea is to define $\Phi(\alpha) = q^F(\mathsf{IPS}^{\varepsilon^\alpha, q^F}_{\sigma,\tau}(\langle\rangle))$ where $\sigma = \mathbb{B}^{\mathbb{N}} \times (\mathbb{B} \times \mathbb{B}^{\mathbb{N}} \to \tau)$ and ε^α, q^F will be suitably defined in terms of α and F respectively. Now, $\mathsf{IPS}^{\varepsilon^\alpha, q^F}(\langle\rangle)$ will be a sequence $\langle A_0^\alpha, A_1^\alpha\rangle \colon \sigma^{\mathbb{N}}$ where $A_0^\alpha \colon (\mathbb{B}^{\mathbb{N}})^{\mathbb{N}}$ and $A_1^\alpha \colon (\mathbb{B} \times \mathbb{B}^{\mathbb{N}} \to \tau)^{\mathbb{N}}$ denote the first and second projections of this sequence. We will define the selection functions ε to ensure that for the first components, $A_0^\alpha(0) = \alpha$ and $A_0^\alpha(n+1) = A_0(n)$, and so in other words

$$\mathsf{IPS}(\langle\rangle) = \langle(\alpha, A_1^\alpha(0)), (\alpha, A_1^\alpha(1)), \ldots, (\alpha, A_1^\alpha(n)), \ldots\rangle.$$

The second components, $A_1^\alpha(n) \colon \mathbb{B} \times \mathbb{B}^{\mathbb{N}} \to \tau$ are more complicated to construct, but the aim is that

$$A_1^\alpha(n) y \beta = \Phi([\alpha](n) * y \mathbin{@} \beta) \text{ if } y = 0 \wedge \alpha(n) = 1.$$

Then, we can simply put $q(A_0, A_1) = F_{\bar{A}_0^\alpha}(\lambda n, y, \beta\ .\ A^\alpha n y \beta)$ where $\bar{A}_0^\alpha$ is the diagonal sequence $\lambda n.A_0^\alpha(n)n = \alpha$. Now, the functional $A_1^\alpha(n)$ is constructed using the selection function $\varepsilon_{[A](n)}$, which has access to the functional

$$p^\alpha_{[A^\alpha](n)} = \lambda(\gamma, g)^\sigma\ .\ q^F(\mathsf{IPS}^{\varepsilon^\alpha, q^F}([A^\alpha](n) * (\gamma, g))).$$

If $\alpha(n) = 0$ we simply define $A_1^\alpha(n) = \mathbf{0}$. On the other hand, if $\alpha(n) = 1$ we set

$$A_1^\alpha(n) y \beta = p^\alpha_{[A^\alpha](n)}((\delta, \mathbf{0})) = q^F(\mathsf{IPS}^{\varepsilon^\alpha, q^F}([A^\alpha](n) * (\delta, \mathbf{0}))).$$

where $\delta = [\alpha](n) * y \mathbin{@} \beta$. We claim that

$$q^F(\mathsf{IPS}^{\varepsilon^\alpha, q^F}([A^\alpha](n) * (\delta, \mathbf{0}))) \overset{(+)}{=} q^F(\mathsf{IPS}^{\varepsilon^\delta, q^F}([A^\delta](n) * (\delta, \mathbf{0}))) \overset{L.2.3}{=} q^F(\mathsf{IPS}^{\varepsilon^\delta, q^F}(\langle\rangle)) = \Phi^F(\delta),$$

the first equality $(+)$ following from a continuity argument, the intuitive idea being that under q^F the sequences $[A^\alpha](n)$ and $[A^\delta](n)$ are treated in the same way whenever $[\alpha](n) = [\delta](n)$.

We now give the construction in full generality for lexicographic orderings induced from arbitrary well-founded relations $<$. In this case we need access to recursors $\mathsf{R}_{(\rho,<),\sigma'}$ for well-founded recursion over $(\rho, <)$ in all finite-types σ', which have the defining equation

$$\mathsf{R}^g_{(\rho,<),\sigma'}(x^\rho) \overset{\sigma'}{=} g_x(\lambda y\ .\ \mathsf{R}^g(y) \text{ if } y < x).$$

We need not assume anything about the strength of $<$ for the proof to be valid, in which case our definability relation becomes $\geq_{\mathsf{T}+\mathsf{R}_<}$, that is, definability in $\mathsf{E}\text{-}\mathsf{HA}^\omega + \mathsf{R}_<$, although if the reader prefers she can assume that $\mathsf{R}_{(\rho,<),\sigma'}$ is already definable in $\mathsf{E}\text{-}\mathsf{HA}^\omega$ in which case we end up with primitive recursive definability in the usual sense. In particular, as we confirm in Section 3.1, for the special case of the BBC functional, the whole construction is done in system $\mathsf{E}\text{-}\mathsf{HA}^\omega$.

Theorem 3.1. $\mathsf{IPS}_{\sigma_{\rho,\tau},\tau} \geq_{\mathsf{T}+\mathsf{R}_{(\rho,<)}} \mathsf{OR}_{(\rho,<),\tau}$ *(instance-wise) over* $\mathsf{Cont} + \mathsf{QF}\text{-}\mathsf{BI}$.

Proof. Suppose that $F \colon \rho^{\mathbb{N}} \times (\mathbb{N} \times \rho \times \rho^{\mathbb{N}} \to \tau) \to \tau$ is some parameter for $\mathsf{OR}_{(\rho,<),\tau}$. Let $\sigma_{\rho,\tau} := \rho^{\mathbb{N}} \times (\rho \times \rho^{\mathbb{N}} \to \tau)$. Define $q^F \colon \sigma^{\mathbb{N}} \to \tau$ by

$$q^F(A_0, A_1) \overset{\tau}{:=} F_{\bar{A}_0}(\lambda n, y, \beta\ .\ A_1 n y \beta)$$

where $A_0 \colon (\rho^{\mathbb{N}})^{\mathbb{N}}$, $A_1 \colon (\rho \times \rho^{\mathbb{N}} \to \tau)^{\mathbb{N}}$ and $\bar{A}_0 := \lambda n.A_0 nn$. For $s \colon \rho^*$, $\gamma \colon \rho^{\mathbb{N}}$ define the selection function $\tilde{\varepsilon}_{s,\gamma} \colon (\sigma \to \tau) \to \sigma$ by

$$\tilde{\varepsilon}_{s,\gamma}(p^{\sigma\to\tau}) :\overset{\sigma}{=} \langle \gamma, f^{s,p}_{\gamma(|s|)} \rangle$$

where the functional $f^{s,p}_x$ is defined using $\mathsf{R}_{(\rho,<),\rho\times\rho^{\mathbb{N}}\to\tau}$ as

$$f^{s,p}_x := \lambda y, \beta \,.\, p(s * y \mathbin{@} \beta, f^{s,p}_y) \text{ if } y < x.$$

Note that if x is minimal with respect to $<$ then $f^{s,p}_x = \mathbf{0}$. Now, given $\alpha \colon \rho^{\mathbb{N}}$ define the dependent selection function $\varepsilon^\alpha \colon \sigma^* \to ((\sigma \to \tau) \to \sigma)$ by

$$\varepsilon^\alpha_t := \tilde{\varepsilon}_{\bar{t}_0,(\alpha * t_0)(|t|)}.$$

where $t_0 \overset{(\rho^{\mathbb{N}})^*}{:=} \lambda n < |t|.\pi_0 tn$ and $(\bar{t}_0) \overset{\rho^*}{:=} \lambda n < |t|.(t_0)nn$. Finally, define

$$\Phi^F_{(\rho,<),\tau}(\alpha) := q^F(\mathsf{IPS}^{\varepsilon^\alpha, q^F}_{\sigma,\tau}(\langle\rangle)).$$

Then $\mathsf{OR}^{\mathsf{ax}}_{(\rho,<),\tau}[\Phi_{(\rho,<),\tau}]$. To prove this, we show that if $A^\alpha := \mathsf{IPS}^{\varepsilon^\alpha, q^F}(\langle\rangle)$ then:

(a) $A^\alpha_0 n = \alpha$ for all n and all α;

(b) $A^\alpha_1 n = \lambda y, \beta \,.\, \Phi^F([\alpha](n) * y \mathbin{@} \beta)$ if $y < \alpha(n)$ for all n and all α.

The result then follows since

$$\begin{aligned}
\Phi^F(\alpha) &= q^F(A^\alpha_0, A^\alpha_1) \\
&= F_{\bar{A}^\alpha_0}(\lambda n, y, \beta \,.\, A^\alpha_1 n y \beta) \\
&\overset{(a),(b)}{=} F_\alpha(\lambda n, y, \beta \,.\, \Phi^F([\alpha](n) * y \mathbin{@} \beta) \text{ if } y < \alpha(n)),
\end{aligned}$$

where the last equality is justified by the substitutions

$$\bar{A}^\alpha_0 = \lambda n.A^\alpha_0 nn \overset{(a)}{=} \lambda n.\alpha(n) = \alpha$$

and

$$\lambda n, y, \beta \,.\, A^\alpha_1 n y \beta \overset{(b)}{=} \lambda n, y, \beta \,.\, \Phi^F([\alpha](n) * y \mathbin{@} \beta) \text{ if } y < \alpha(n).$$

Therefore it remains to prove (a) and (b). First, note that by Theorem 2.4, for all n, α we have

$$\begin{aligned}
A^\alpha n &= \varepsilon^\alpha_{[A^\alpha](n)}(p^\alpha_{[A^\alpha](n)}) \\
p^\alpha_{[A^\alpha](n)}(A^\alpha n) &= q^F(A^\alpha)
\end{aligned} \tag{6}$$

where $p^\alpha_{[A^\alpha](n)} := \lambda z \,.\, q^F(\mathsf{IPS}^{\varepsilon^\alpha, q^F}([A^\alpha](n) * z))$. We prove (a) by induction, since for arbitrary α we have, by (6):

$$\begin{aligned}
A^\alpha_0 0 &= \varepsilon^\alpha_{\langle\rangle}(p^\alpha_{\langle\rangle})_0 = \tilde{\varepsilon}_{\langle\rangle,\alpha}(p^\alpha_{\langle\rangle})_0 = \alpha \\
A^\alpha_0(n+1) &= \varepsilon^\alpha_{[A^\alpha](n+1)}(p^\alpha_{[A^\alpha](n+1)})_0 \overset{I.H.}{=} \tilde{\varepsilon}_{[\alpha](n+1),\alpha}(p^\alpha_{[A^\alpha](n+1)})_0 = \alpha.
\end{aligned}$$

In particular note that

$$A^\alpha n = \varepsilon^\alpha_{[A^\alpha](n)}(p^\alpha_{[A^\alpha](n)}) = \tilde{\varepsilon}_{[\alpha](n),\alpha}(p^\alpha_{[A^\alpha](n)})$$

for all α, n. For (b), then, we see that

$$\begin{aligned}
A_1^\alpha n y \beta &= \tilde{\varepsilon}_{[\alpha](n),\alpha}(p^\alpha_{[A^\alpha](n)})_1 y\beta \\
&= f^{[\alpha](n),p^\alpha_{[A^\alpha](n)}}_{\alpha(n)} y\beta \\
&= p^\alpha_{[A^\alpha](n)}([\alpha](n) * y \mathbin{@} \beta, f_y^{[\alpha](n),p^\alpha_{[A^\alpha](n)}}) \text{ if } y < \alpha(n) \\
&\overset{(*)}{=} p^\alpha_{[A^\alpha](n)}(\delta, f^{[\delta](n),p^\alpha_{[A^\alpha](n)}}_{\delta(n)}) \text{ if } y < \alpha(n) \\
&\overset{(+)}{=} p^\delta_{[A^\delta](n)}(\delta, f^{[\delta](n),p^\delta_{[A^\delta](n)}}_{\delta(n)}) \text{ if } y < \alpha(n) \\
&= p^\delta_{[A^\delta](n)}(\tilde{\varepsilon}_{[\delta](n),\delta}(p^\delta_{[A^\delta](n)})) \text{ if } y < \alpha(n) \\
&= p^\delta_{[A^\delta](n)}(A^\delta n) \text{ if } y < \alpha(n) \\
&\overset{(6)}{=} q^F(A^\delta) \text{ if } y < \alpha(n) \\
&\overset{(*)}{=} \Phi^F([\alpha](n) * y \mathbin{@} \beta) \text{ if } y < \alpha(n)
\end{aligned}$$

where for $(*)$ we have simply defined $\delta := [\alpha](n) * y \mathbin{@} \beta$. All that remains is to prove $(+)$ which follows from the claim that $p^\alpha_{[A^\alpha](n)} = p^\delta_{[A^\delta](n)}$. This claim is proved using a fairly straightforward bar induction argument, and is given as Lemma A.1. □

3.1 A bar recursive construction of BBC

An immediate consequence of the previous result is the following.

Corollary 3.2. $\mathsf{IPS}_{\sigma_{\bar\rho,\tau},\tau} \geq_\mathsf{T} \mathsf{UR}_{\rho,\tau} \geq_\mathsf{T} \mathsf{BBC}_{\rho,\tau}$ *(instance-wise) over* $\mathsf{Cont} + \mathsf{QF\text{-}BI}$.

Proof. By Theorem 2.12, $\mathsf{UR}_{\rho,\tau}$ is definable from an instance of $\mathsf{OR}_{(\bar\rho,<),\tau}$ on the well-founded relation $\bot < x$. Clearly recursion over $<$ is definable in system $\mathsf{E\text{-}HA}^\omega$, therefore the result follows directly from Theorem 3.1. □

By inspecting the proofs of Theorems 2.12 and 3.1, we can give the construction of $\mathsf{BBC}_{\rho,\tau}$ from $\mathsf{IPS}_{\sigma_{\bar\rho\,\tau},\tau}$ directly. Given parameters $r\colon \rho^\mathbb{N} \to \tau$ and $\delta_n\colon (\rho \to \tau) \to \rho$ for $\mathsf{BBC}_{\rho,\tau}$ define $q^{\delta,r}\colon \sigma^\mathbb{N} \to \tau$ and $\varepsilon^u\colon \sigma^* \to ((\sigma \to \tau) \to \sigma)$ (where $\sigma \equiv \sigma_{\bar\rho,\tau} :\equiv \bar\rho^\mathbb{N} \times (\bar\rho \times \bar\rho^\mathbb{N} \to \tau)$) by

$$q^{\delta,r}(A_0, A_1) :\overset{\tau}{=} r(\bar{A}_0 \mathbin{@} \lambda n\,.\,\delta_n(\lambda x^\rho\,.\,A_1 n x \bar{A}_0))$$

$$\varepsilon^u_t(p^{\sigma\to\tau}) :\overset{\sigma}{=} \begin{cases} \langle (u * t_0)(|t|), f^{\bar{t}_0,p}\rangle & \text{if } |t| \notin \mathrm{dom}((u * t_0)|t|) \\ \langle (u * t_0)(|t|), \mathbf{0}\rangle & \text{otherwise.} \end{cases}$$

where $f^{s,p} := \lambda y, v\,.\,p(s * y \mathbin{@} v, \mathbf{0})$ if $y \neq \bot$. Then $\Phi^{\delta,r}_{\rho,\tau}(u) := q^{\delta,r}(\mathsf{IPS}^{\varepsilon^u,q^{\delta,r}}_{\sigma,\tau}(\langle\rangle))$ satisfies the defining equation of $r(\mathsf{BBC}^{\delta,r}(u))$ (which in turn easily defines $\mathsf{BBC}^{\delta,r}$). Using the same notation as in Theorem 3.1, we have

$$\begin{aligned}
\Phi^{\delta,r}(u) &= r(\bar{A}^u_0 \mathbin{@} \lambda n\,.\,\delta_n(\lambda x\,.\,A^u_1 n x \bar{A}^u_0)) \\
&\overset{(a)}{=} r(u \mathbin{@} \lambda n\,.\,\delta_n(\lambda x\,.\,A^u_1 n x u)) \\
&\overset{(b)}{=} r(u \mathbin{@} \lambda n\,.\,\delta_n(\lambda x\,.\,\Phi([u](n) * x \mathbin{@} u))) \\
&= r(u \mathbin{@} \lambda n\,.\,\delta_n(\lambda x\,.\,\Phi(u^x_n)))
\end{aligned}$$

where (a) follows since $A_0^u n = u$ for all n and the equality (b) is justified by

$$\begin{aligned} A_1^u nxu &\overset{(i)}{=} f^{[u](n),p^u_{[A^u](n)}} xu \\ &= p^u_{[A^u](n)}([u](n) * x \mathbin{@} u, \mathbf{0}) \\ &\overset{(+)}{=} p^w_{[A^w](n)}(w, \mathbf{0}) \\ &\overset{(i)}{=} p^w_{[A^w](n)}(A^w n) \\ &\overset{Th.2.4}{=} q^{\varepsilon,q}(A^w) \\ &= \Phi(u^x_n) \end{aligned}$$

where we use the abbreviation $w := [u](n)*(1, x) \mathbin{@} u = u^x_n$. The steps labelled (i) follow from definition of $\tilde{\varepsilon}$ and the fact that $n \notin \mathrm{dom}(u)$ and $n \in \mathrm{dom}(w)$ respectively. Step $(+)$ follows from Lemma 2.3.

4 Defining IPS from BBC

By combining our work of the previous section with the existing results outlined in Section 2 we obtain the sequence

$$\mathsf{ips} \geq_{\mathsf{T}} \mathsf{IPS} \geq_{\mathsf{T}} \mathsf{OR} \geq_{\mathsf{T}} \mathsf{UR} \geq_{\mathsf{T}} \mathsf{BBC}.$$

We now close the circle by proving that $\mathsf{BBC} \geq_{\mathsf{T}} \mathsf{ips}$. As in the previous section, we begin with an informal discussion of our construction.

One obvious problem with building ips from BBC is the issue of dependency: as we discussed in Section 2.3, the BBC functional returns a sequence in which each element is computed independently of the others, ips computes its output sequentially. We solve this problem by simulating $\mathsf{ips}_{\rho,\tau}$ with $\mathsf{BBC}_{\bar{\rho}^{\mathbb{N}},\tau}$. This instance of BBC returns a sequence of *partial sequences* i.e. an infinite matrix over the type $\bar{\rho}$.

Our idea is to compute $\mathsf{ips}(\langle\rangle)$ as just the first column of this matrix, i.e. $\mathsf{BBC}(\emptyset)(0)$ (we simply ignore the columns $\mathsf{BBC}(\emptyset)(n)$ for $n > 0$). Now, while ips is computed in the first column, for the recursive calls of ips we shift the computation to the next column, so to compute $\mathsf{ips}(t_n * x)$ we look at $\mathsf{BBC}(\langle t_n * x\rangle)(1)$ where $\langle t_n * x\rangle$ denotes a partial function $\mathbb{N} \to \bar{\rho}^{\mathbb{N}}$ given by $\langle t_n * x\rangle(0) = t_n * x * \langle \bot, \bot, \ldots$ and $\langle t_n * x\rangle(n)$ is undefined (on the higher level) for $n > 0$. Note that we only appeal to what BBC outputs in column 1 during a recursive call in the computation of column 0. In order to carry out this subcomputation in column 1, we again move to the next column whenever we need to make a recursive call, so we only look at what BBC outputs in column 2 during a nested recursive call for column 0 followed by column 1, and so on. The basic idea is that ips is computed using only part of the whole BBC functional: in order to compute $\mathsf{BBC}(\emptyset)$ we only need to know the value of $\mathsf{BBC}(\langle u, \bot, \bot, \ldots\rangle)(1)$ for various partial functions u, for which in turn we only need to know the value of $\mathsf{BBC}(\langle u, v, \bot, \bot, \ldots\rangle)(2)$ for various v and so on. Thus the whole computation is contained inside just one particular tree of recursive calls in BBC.

Without further ado, we now give the general construction. First we need some notation. Given a finite sequence $s\colon \rho^*$ let $\overline{s}\colon \bar{\rho}^{\mathbb{N}}$ denote its embedding as a partial function i.e.

$$\overline{s}(n) := \begin{cases} s_n & \text{if } n < |s| \\ \bot & \text{otherwise.} \end{cases}$$

In the following we will often, however, omit this notation and just write s when it is obvious that we're treating it as a partial function.

Theorem 4.1. $\mathsf{BBC}_{\bar\rho^{\mathbb{N}},\tau} \geq_{\mathsf{T}} \mathsf{ips}_{\rho,\tau}$ *(instance-wise) over* $\mathsf{Cont} + \mathsf{QF\text{-}BI}$.

Proof. Suppose we are given parameters $\varepsilon_n \colon ((\rho \to \tau) \to \rho)$ and $q \colon \rho^{\mathbb{N}} \to \tau$ for $\mathsf{ips}_{\rho,\tau}$. First, define $\tilde q \colon (\bar\rho^{\mathbb{N}})^{\mathbb{N}} \to \tau$ by

$$\tilde q(\gamma) := q(d(\gamma))$$

where $d \colon (\bar\rho^{\mathbb{N}})^{\mathbb{N}} \to \rho^{\mathbb{N}}$ is constructed using a bounded search as

$$d(\gamma)(n) \overset{\rho}{:=} \begin{cases} \gamma m n & \text{for least } m \leq n \text{ with } n \in \mathrm{dom}(\gamma m) \\ \mathbf{0} & \text{if no such } m \text{ exists.} \end{cases}$$

Second, using course-of-values recursion of type ρ define the family of selection functions $\tilde\varepsilon_n \colon (\bar\rho^{\mathbb{N}} \to \tau) \to \bar\rho^{\mathbb{N}}$ by $\tilde\varepsilon_n := \tilde\varepsilon$ where

$$\tilde\varepsilon(p^{\bar\rho^{\mathbb{N}} \to \tau}) \overset{\bar\rho^{\mathbb{N}}}{:=} \lambda k \,.\, \varepsilon_k(\lambda x^{\rho} \,.\, p(\overline{\tilde t_k * x}))$$

where $\tilde t_k :=_{\rho^*} [\tilde\varepsilon(p)](k)$. Note that $\tilde\varepsilon(p)$ is a total sequence (though officially embedding in the type $\bar\rho^{\mathbb{N}}$) and hence $\tilde t_k$ can be viewed as an element of type ρ^*, even though each recursive call on p is a partial sequence. Finally, define

$$\Phi^{\varepsilon,q}_{\rho,\tau}(s^{\rho^*}) \overset{\rho^{\mathbb{N}}}{:=} \begin{cases} d(\mathsf{BBC}^{\tilde\varepsilon,\tilde q}_{\bar\rho^{\mathbb{N}},\tau}(\emptyset)) & \text{if } |s| = 0 \\ d(\mathsf{BBC}^{\tilde\varepsilon,\tilde q}_{\bar\rho^{\mathbb{N}},\tau}(\overline{\langle \bar s \rangle})) & \text{otherwise.} \end{cases}$$

We prove that $\mathsf{ips}^{\mathsf{ax}}_{\rho,\tau}[\Phi_{\rho,\tau}]$. To cut down on excessive syntax, given a sequence $\langle u_0, \ldots, u_{k-1} \rangle$ of finite sequences $u_i \colon \rho^*$, we shall write $\mathsf{BBC}(\langle u_0, \ldots, u_{k-1} \rangle)$ for BBC applied to the partial array $\overline{\langle \overline{u}_0, \ldots, \overline{u}_{k-1} \rangle}$, whose type is *partial* sequence $\mathbb{N} \to \bar\rho^{\mathbb{N}}$, so in our shorthand we define $\mathsf{ips}(s) := d(\mathsf{BBC}(\langle s \rangle))$ for $|s| > 0$. Note that the input type of BBC has undefined elements on two levels: undefined elements of the argument correspond to undefined 'columns', while defined columns of type $\bar\rho^{\mathbb{N}}$ may themselves have undefined entries.

To verify our construction, first observe that for $|s| = 0$ we have

$$\begin{aligned} \Phi(\langle\rangle) &\overset{(i)}{=} \lambda n \,.\, \tilde\varepsilon_0(\underbrace{\lambda f \,.\, \tilde q(\mathsf{BBC}(\emptyset^f_0))}_{p})(n) \\ &= \lambda n \,.\, \varepsilon_n(\lambda x \,.\, \tilde q(\mathsf{BBC}(\emptyset^{\overline{\tilde t_n * x}}_0))) \\ &= \lambda n \,.\, \varepsilon_n(\lambda x \,.\, q(d(\mathsf{BBC}(\langle \tilde t_n * x \rangle)))) \\ &\overset{(ii)}{=} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, q(\Phi(t_n * x))) \end{aligned}$$

where for (ii) we have $\tilde t_n = [\tilde\varepsilon_0(p)](n) = [\Phi(\langle\rangle)](n) = t_n$. Here (i) follows because $n \in \mathrm{dom}(\mathsf{BBC}(\emptyset)(0)) = \mathrm{dom}(\tilde\varepsilon_0(p))$ for for all n and therefore $d(\mathsf{BBC}(\emptyset))(n) = \mathsf{BBC}(\emptyset)(0)(n) = \tilde\varepsilon_0(p)(n)$. Similarly, for $|s| > 0$ we have

$$\begin{aligned} \Phi(s) &\overset{(i)}{=} s \mathbin{@} \lambda n \,.\, \tilde\varepsilon_1(\underbrace{\lambda f \,.\, \tilde q(\mathsf{BBC}(\langle s \rangle^f_1))}_{p})(n) \\ &= s \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, \tilde q(\mathsf{BBC}(\langle s \rangle^{\overline{\tilde t_n * x}}_1))) \\ &= s \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, q(d(\mathsf{BBC}(\langle s, \tilde t_n * x \rangle)))) \\ &\overset{(+)}{=} s \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, q(d(\mathsf{BBC}(\langle s \mathbin{@} (\tilde t_n * x) \rangle)))) \\ &= s \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, q(\Phi(s \mathbin{@} (\tilde t_n * x)))) \\ &\overset{(ii)}{=} s \mathbin{@} \lambda n \,.\, \varepsilon_n(\lambda x \,.\, q(\Phi(t_n * x))) \end{aligned}$$

where for (ii) we have (for $n \geq |s|$)

$$\begin{aligned} s \mathbin{@} (\tilde{t}_n * x) &= s \mathbin{@} ([\tilde{\varepsilon}_1(p)](n) * x) \\ &= s * \langle \tilde{\varepsilon}_1(p)(|s|), \ldots, \tilde{\varepsilon}_1(p)(n-1) \rangle * x \\ &= [\Phi(s)](n) * x \\ &= t_n * x. \end{aligned}$$

Again, (i) follows because $n \in \mathrm{dom}(\mathsf{BBC}(\langle s \rangle)(0)) = \mathrm{dom}(\overline{s})$ iff $n < |s|$ and $n \in \mathrm{dom}(\mathsf{BBC}(\langle s \rangle)(1)) = \mathrm{dom}(\tilde{\varepsilon}_1(p))$ otherwise. Therefore

$$d(\mathsf{BBC}(\langle s \rangle))(n) = \begin{cases} s_n & \text{if } n < |s| \\ \tilde{\varepsilon}_1(p)(n) & \text{otherwise.} \end{cases}$$

All that remains is to justify $(+)$, which follows from the claim that

$$\tilde{q}(\mathsf{BBC}(\langle s, \tilde{t}_n * x \rangle)) = \tilde{q}(\mathsf{BBC}(\langle s \mathbin{@} (\tilde{t}_n * x) \rangle))$$

We prove this claim via another fairly routine continuity argument in Lemma A.2 below. □

5 Concluding remarks

Together, our results from Sections 3 and 4 establish the following:

Theorem 5.1. *Each of open recursion, update recursion and the BBC functional are primitive recursively equivalent to* PCF, *or alternatively modified bar recursion, over* Cont + QF-BI

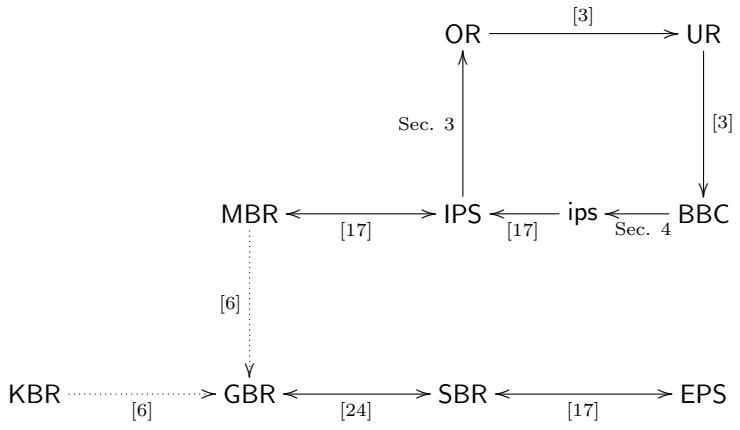

Figure 1: Summary of definability results

In Figure 1 we provide a simple map of the main definability results mentioned in this paper (the dotted arrows representing directions whose converses are false). Our diagram is a simplification of all the known definability results which concern extensions of system T (see [8, 5, 17] for example), although most variants of bar recursion found in the context of proof theory nevertheless fall into one of

the two main equivalence classes represented by either IPS or GBR. The results of this paper unify the well-known recursive extensions of system T attached to proof interpretations, showing the equivalence of two very different styles of recursion that up until now have mostly been treated separately.

Moreover, our proofs achieve strictly more that the equivalence of bar recursion and open recursion, in the sense that we give explicit and fairly elementary instance-wise constructions in both directions, which can be verified in an entirely standard and relatively weak theory which is validated by a variety of constructive models of system T, such as the Kleene-Kreisel continuous functionals $\mathcal{C}^\omega$.

Theorem 5.1 allows us to lift many of the known properties of IPS and MBR to the open recursive functionals. In particular, we obtain a new proof that open recursion and BBC exists in $\mathcal{C}^\omega$ - namely as a consequence of the existence of IPS - without having to resort to open induction. Also, by Theorem 2.10 we obtain the following:

Corollary 5.2. *None of open induction, update induction nor the BBC functional are S1-S9 computable over the total continuous functionals.*

Corollary 5.2 can actually be viewed as a negative result, as it confirms that while the BBC functional and open induction have both been proposed in order to obtain programs from classical proofs that are more direct or efficient in some way, nevertheless the Dialectica interpretation and Spector's bar recursion remain optimal in the sense that they produce computable witnesses from proofs in classical analysis. An obvious challenge for future research is to construct 'explicitly well-founded' analogues to these forms of recursion that are both direct and computable.

We leave open the possibility that our results can be improved at the level of types, and in particular do not claim to have provided any insight into how the specific instances of recursion used to interpret choice principles compare in terms of efficiency or computational complexity. We also leave open the question of whether or not we could weaken the theory $\mathsf{Cont} + \mathsf{QF\text{-}BI}$ so that our equivalences would hold over non-continuous structures such as the majorizable functionals $\mathcal{M}^\omega$.

It would be interesting to develop a semantic framework along the lines of [14] in which the known extensions of system T can be compared, and also to explore the construction of more exotic variants of bar recursion like KBR in order to exhibit a richer variety of extensions of T and learn more about them. For now, however, we have succeeded in relating two well-known and important extensions of the primitive recursive functionals.

Acknowledgments. Earlier versions of this work were read by Martin Hyland, Ulrich Kohlenbach and Paulo Oliva, whose helpful and insightful comments have led to a much improved presentation. The author is also grateful to Ulrich Berger and Martín Escardó for several interesting discussions on the topic of this paper. The main part of this research was carried out while the author was in receipt of an EPSRC doctoral training grant, and the final paper written during a LabEx CARMIN research fellowship at IHÉS.

A Omitted proofs

Lemma A.1. *Let A^α and p_t^α be defined as in Theorem 3.1 as*

$$A^\alpha = \mathsf{IPS}^{\varepsilon^\alpha, q^F}(\langle\rangle)$$
$$p_t^\alpha = \lambda z \,.\, q^F(\mathsf{IPS}^{\varepsilon^\alpha, q^F}(t * z))$$

Then if $[\alpha](n) = [\delta](n)$ then $p^\alpha_{[A^\alpha](n)} = p^\delta_{[A^\delta](n)}$ for arbitrary α, δ and n, provably over $\mathsf{Cont} + \mathsf{QF\text{-}BI}$.

Proof. We use induction on n, with an auxiliary bar induction needed to establish the main induction step. Suppose that $[\alpha](n) = [\delta](n)$, and assume for the main induction hypothesis that

$$\forall m < n, \alpha', \delta'([\alpha'](m) = [\delta'](m) \to p^{\alpha'}_{[A^{\alpha'}](m)} = p^{\delta'}_{[A^{\delta'}](m)}).$$

We want to prove that $p^{\alpha}_{[A^{\alpha}](n)} z = p^{\delta}_{[A^{\delta}](n)} z$ for all $z \colon \sigma$, which we do using an auxiliary $\mathsf{QF\text{-}BI}$ on the formula

$$P(t^{\sigma^*}) :\equiv q^F(\mathsf{IPS}^{\varepsilon^\alpha}([A^\alpha](n) * z * t)) \stackrel{\tau}{=} q^F(\mathsf{IPS}^{\varepsilon^\delta}([A^\delta](n) * z * t))$$

(with $S(t) = \mathsf{true}$), the result following from $P(\langle\rangle)$. The main induction hypothesis is used to verify that $A^\alpha_1 m = A^\delta_1 m$ for all $m < n$. To see this, note that

$$\begin{aligned}
A^\alpha_1 m &= \tilde{\varepsilon}_{[\alpha](m),\alpha}(p^\alpha_{[A^\alpha](m)})_1 \\
&= f^{[\alpha](m), p^\alpha_{[A^\alpha](m)}}_{\alpha(m)} \\
&\stackrel{I.H.}{=} f^{[\delta](m), p^\delta_{[A^\delta](m)}}_{\delta(m)} \\
&= \tilde{\varepsilon}_{[\delta](m),\delta}(p^\delta_{[A^\delta](m)})_1 \\
&= A^\delta_1 m.
\end{aligned}$$

This enables us to establish the bar for the auxiliary bar induction: For an arbitrary infinite sequence $B \colon \sigma^{\mathbb{N}}$ let N be the point of continuity of $q^F_{[A^\alpha](n)*z}$ on B. Then

$$\begin{aligned}
q^F(\mathsf{IPS}^{\varepsilon^\alpha}([A^\alpha](n) * z * [B](N))) &= q^F([A^\alpha](n) * z * [B](N) \mathbin{@} \mathsf{IPS}^{\varepsilon^\alpha}([A^\alpha](n) * z * [B](N))) \\
&\stackrel{\mathsf{Cont}}{=} q^F([A^\alpha](n) * z * [B](N) \mathbin{@} \mathsf{IPS}^{\varepsilon^\delta}([A^\delta](n) * z * [B](N))) \\
&\stackrel{(i)}{=} F_{\overline{[A^\alpha_0](n) * z_0 * [B_0](N) \mathbin{@} C_0}}([A^\alpha_1](n) * z_1 * [B_1](N) \mathbin{@} C_1) \\
&\stackrel{(ii)}{=} F_{\overline{[A^\delta_0](n) * z_0 * [B_0](N) \mathbin{@} C_0}}([A^\delta_1](n) * z_1 * [B_1](N) \mathbin{@} C_1) \\
&= q^F([A^\delta](n) * z * [B](N) \mathbin{@} \mathsf{IPS}^{\varepsilon^\delta}([A^\delta](n) * z * [B](N))) \\
&= q^F(\mathsf{IPS}^{\varepsilon^\delta}([A^\delta](n) * z * [B](N)))
\end{aligned}$$

Here for (i) we simply use the abbreviation $C := \mathsf{IPS}^{\varepsilon^\delta}([A^\delta](n) * z * [B](N))$ and definition of q^F and (ii) follows by the fact that $[A^\alpha_1](n) = [A^\delta_1](n)$ and $\overline{[A^\alpha_0](n)} = [\alpha](n) = [\delta](n) = \overline{[A^\delta_0](n)}$. Now the bar induction step is straightforward. Assuming that $\forall z' P(t * z')$ we have

$$\begin{aligned}
q^F(\mathsf{IPS}^{\varepsilon^\alpha}([A^\alpha](n) * z * t)) &= q^F(\mathsf{IPS}^{\varepsilon^\alpha}([A^\alpha](n) * z * t * a)) \\
&\stackrel{B.I.H}{=} q^F(\mathsf{IPS}^{\varepsilon^\delta}([A^\delta](n) * z * t * a)) \\
&\stackrel{(*)}{=} q^F(\mathsf{IPS}^{\varepsilon^\delta}([A^\delta](n) * z * t * a')) \\
&= q^F(\mathsf{IPS}^{\varepsilon^\delta}([A^\delta](n) * z * t))
\end{aligned}$$

where for $(*)$ we have

$$\begin{aligned}
a &= \varepsilon^{\alpha}_{[A^\alpha](n)*z*t}(\lambda z' \,.\, q^F(\mathsf{IPS}^{\varepsilon^\alpha}([A^\alpha](n) * z * t * z'))) \\
&\overset{B.I.H.}{=} \varepsilon^{\alpha}_{[A^\alpha](n)*z*t}(\lambda z' \,.\, q^F(\mathsf{IPS}^{\varepsilon^\delta}([A^\delta](n) * z * t * z'))) \\
&= \tilde{\varepsilon}_{\overline{[A^\alpha_0](n)*z_0*t_0},t'}(\lambda z' \,.\, q^F(\mathsf{IPS}^{\varepsilon^\delta}([A^\delta](n) * z * t * z'))) \\
&= \tilde{\varepsilon}_{\overline{[A^\delta_0](n)*z_0*t_0},t'}(\lambda z' \,.\, q^F(\mathsf{IPS}^{\varepsilon^\delta}([A^\delta](n) * z * t * z'))) \\
&= \varepsilon^{\delta}_{[A^\delta](n)*z*t}(\lambda z' \,.\, q^F(\mathsf{IPS}^{\varepsilon^\delta}([A^\delta](n) * z * t * z'))) \\
&= a'
\end{aligned}$$

where t' denotes tail element of the non-empty sequence $z * t$. This completes the bar induction step $\forall t(\forall z' P(t*z') \to P(t))$ and along with the preceding continuity argument establishing $\forall B \exists N P([B](N))$ we obtain by bar induction $P(\langle\rangle)$ and hence $p^{\alpha}_{[A^\alpha](n)} = p^{\delta}_{[A^\delta](n)}$, assuming throughout the *main*, *normal* induction hypothesis. So in turn this completes the main induction step, thereby obtaining by normal induction $\forall n, \alpha, \delta([\alpha](n) = [\delta](n) \to p^{\alpha}_{[A^\alpha](n)} = p^{\delta}_{[A^\delta](n)})$, so we're done. □

Lemma A.2. *Whenever* $0 < |s| < |t|$*, we have* $\tilde{q}(\mathsf{BBC}^{\tilde{\varepsilon},\tilde{q}}(\langle s, t\rangle)) = \tilde{q}(\mathsf{BBC}^{\tilde{\varepsilon},\tilde{q}}(\langle s \mathbin{@} t\rangle))$*, provably in* $\mathsf{Cont} + \mathsf{QF\text{-}BI}$.

Proof. Recall that in the argument of BBC, the terms $\langle s, t\rangle$ and $\langle s * t\rangle$ are shorthand for $\overline{\langle \overline{s}, \overline{t}\rangle}$ and $\overline{\langle \overline{s \mathbin{@} t}\rangle}$ respectively, so written informally we're asserting that

$$\tilde{q}(\mathsf{BBC}\begin{pmatrix} s_0 & t_0 \\ \vdots & \vdots \\ s_i & t_i \\ \bot & t_{i+1} \\ \vdots & \vdots \\ & t_j \\ & \bot \\ & \vdots \end{pmatrix}) = \tilde{q}(\mathsf{BBC}\begin{pmatrix} s_0 \\ \vdots \\ s_i \\ t_{i+1} \\ \vdots \\ t_j \\ \bot \\ \vdots \end{pmatrix})$$

where all other columns are undefined (on the higher level). We prove this using bar induction on sequences $u\colon (\rho^*)^*$ relativised to those consisting only of non-empty sequences i.e. $S(u) :\equiv \forall i < |u|(|u_i| > 0)$. For our main predicate P, first define $u^\bullet, u^\circ\colon (\rho^*)^*$ with $|u^\bullet| = |u^\circ| = |u|$ recursively in BBC as

$$\begin{aligned}
u^\bullet_i &:= [\tilde{\varepsilon}(\lambda f \,.\, \tilde{q}(\mathsf{BBC}(\langle s, t, u^\bullet_0, \dots, u^\bullet_{i-1}\rangle^f_{i+2})))](k_i) * u_i \\
u^\circ_i &:= [\tilde{\varepsilon}(\lambda f \,.\, \tilde{q}(\mathsf{BBC}(\langle s \mathbin{@} t, u^\circ_0, \dots, u^\circ_{i-1}\rangle^f_{i+1})))](k_i) * u_i
\end{aligned}$$

where $k_i := |t| + \sum_{j=0}^{i-1} |u_j|$. The purpose of these constructions is actually to simplify the bar induction: the first k_i entries of $u^\bullet_i$ (and analogously u°_i) represent 'dummy' variables that would be computed by $\mathsf{BBC}(\langle s, t, u^\bullet_i, \dots, u^\bullet_{i-1}\rangle)$ in the $(i+2)$th column but ignored by the outcome functional $\tilde{q}$. We carry out bar induction on the quantifier-free predicate

$$P(u) :\equiv \tilde{q}(\mathsf{BBC}(\langle s, t\rangle * u^\bullet)) = \tilde{q}(\mathsf{BBC}(\langle s \mathbin{@} t\rangle * u^\circ)).$$

The lemma then follows from $P(\langle\rangle)$. First, to establish the bar, suppose that $\alpha\colon (\rho^*)^{\mathbb{N}}$ is an infinite sequence of non-empty sequences, and let N be a point of continuity of q_{s*t} on the sequence $\tilde{\alpha}\colon \rho^{\mathbb{N}}$ defined by

$$\tilde{\alpha}(n) = \alpha_0 * \alpha_1 * \alpha_2 * \dots.$$

Because the α_i are non-empty, $[\tilde{\alpha}](N)$ is a prefix of $\alpha_0 * \ldots * \alpha_{N-1}$, and therefore by unwinding the definition of d we see that $(s \mathbin{@} t) * [\tilde{\alpha}](N)$ is a prefix of both $d(\mathsf{BBC}(\langle s,t\rangle * [\alpha](N)^\bullet))$ and $d(\mathsf{BBC}(\langle s \mathbin{@} t\rangle * [\alpha](N)^\circ))$. Thus by continuity we have

$$\tilde{q}(\mathsf{BBC}(\langle s,t\rangle * [\alpha](N)^\bullet)) = \tilde{q}(\mathsf{BBC}(\langle s \mathbin{@} t\rangle * [\alpha](N)^\circ)).$$

For the bar induction step, take an arbitrary sequence u of non-empty finite sequences. Again, by definition of d we have

$$d(\mathsf{BBC}(\langle s,t\rangle * u^\bullet)) = (s \mathbin{@} t) * u_0 * \ldots u_{|u|-1} * \beta$$

where

$$\begin{aligned}
\beta(n) &= \tilde{\varepsilon}_{|u|+2}(\underbrace{\lambda f \,.\, \tilde{q}(\mathsf{BBC}((\langle s,t\rangle * u^\bullet)^f_{|u|+2}))}_{p})(k_{|u|}+n) \\
&= \varepsilon_{k_{|u|}+n}(\lambda x \,.\, \tilde{q}(\mathsf{BBC}((\langle s,t\rangle * u^\bullet)^{\overline{\tilde{t}_{k_{|u|}+n} *_\rho x}}_{|u|+2}))) \\
&= \varepsilon_{k_{|u|}+n}(\lambda x \,.\, \tilde{q}(\mathsf{BBC}(\langle s,t\rangle * u^\bullet * \langle \tilde{t}_{k_{|u|}+n} *_\rho x\rangle))) \\
&\overset{(i)}{=} \varepsilon_{k_{|u|}+n}(\lambda x \,.\, \tilde{q}(\mathsf{BBC}(\langle s,t\rangle * (u * ([\beta](n) *_\rho x))^\bullet))).
\end{aligned}$$

Note that these formulas feature sequence concatenation of both type ρ and ρ^*, which we have highlighted. For (i) we clearly have $u_i^\bullet = (u * ([\beta](n) *_\rho x))^\bullet_i$ for $i < |u|$ since the operation $(-)^\bullet$ is recursive on sequences, and for $i = |u|$ we have

$$\begin{aligned}
\tilde{t}_{k_{|u|}+n} *_\rho x &= \langle \tilde{\varepsilon}_{|u|+2}(p)(0), \ldots, \tilde{\varepsilon}_{|u|+2}(p)(k_{|u|}-1)\rangle * \langle \tilde{\varepsilon}_{|u|+2}(p)(k_{|u|}), \ldots, \tilde{\varepsilon}_{|u|+2}(p)(k_{|u|}+n-1)\rangle * x \\
&= [\tilde{\varepsilon}(\lambda f \,.\, \tilde{q}(\mathsf{BBC}((\langle s,t\rangle * u^\bullet)^f_{|u|+2})))](k_{|u|}) * ([\beta](n) * x) \\
&= (u * ([\beta](n) * x))^\bullet_{|u|}.
\end{aligned}$$

By an analogous argument we have

$$d(\mathsf{BBC}(\langle s \mathbin{@} t\rangle * u^\circ)) = (s \mathbin{@} t) * u_0 * \ldots * u_{|u|-1} * \beta'$$

where

$$\beta'(n) = \varepsilon_{k_{|u|}+n}(\lambda x \,.\, \tilde{q}(\mathsf{BBC}(\langle s \mathbin{@} t\rangle * (u * ([\beta'](n) *_\rho x))^\circ))).$$

We now use the bar induction hypothesis to prove that $\beta(n) = \beta'(n)$ for all n. We use this main bar induction hypothesis to prove the induction step in an auxiliary course-of-values induction over $\mathbb{N}$. Suppose as a course-of-values induction hypothesis that $\beta(m) = \beta'(m)$ for all $m < n$. Then

$$\begin{aligned}
\beta(n) &= \varepsilon_{k_{|u|}+n}(\lambda x \,.\, \tilde{q}(\mathsf{BBC}(\langle s,t\rangle * (u * ([\beta](n) *_\rho x))^\bullet))) \\
&\overset{B.I.H.}{=} \varepsilon_{k_{|u|}+n}(\lambda x \,.\, \tilde{q}(\mathsf{BBC}(\langle s \mathbin{@} t\rangle * (u * ([\beta](n) *_\rho x))^\circ))) \\
&\overset{I.H.}{=} \varepsilon_{k_{|u|}+n}(\lambda x \,.\, \tilde{q}(\mathsf{BBC}(\langle s \mathbin{@} t\rangle * (u * ([\beta'](n) *_\rho x))^\circ))) \\
&\beta'(n).
\end{aligned}$$

Therefore $\beta = \beta'$, and we finally have

$$\begin{aligned}
\tilde{q}(\mathsf{BBC}(\langle s,t\rangle * u^\bullet)) &= q((s \mathbin{@} t) * u_0 * \ldots * u_{|u|-1} * \beta) \\
&= q((s \mathbin{@} t) * u_0 * \ldots * u_{|u|-1} * \beta') \\
&= \tilde{q}(\mathsf{BBC}(\langle s \mathbin{@} t\rangle * u^\circ))
\end{aligned}$$

This completes the bar induction. □

References

[1] S. Berardi, M. Bezem, and T. Coquand. On the computational content of the axiom of choice. *Journal of Symbolic Logic*, 63(2):600–622, 1998.

[2] U. Berger. The Berardi-Bezem-Coquand functional in a domain-theoretic setting. Unpublished results, available from author's webpage at http://www.cs.swan.ac.uk/∼csulrich/recent-papers.html, 2002.

[3] U. Berger. A computational interpretation of open induction. In F. Titsworth, editor, *Proceedings of the Nineteenth Annual IEEE Symposium on Logic in Computer Science*, pages 326–334. IEEE Computer Society, 2004.

[4] U. Berger. Strong normalization for applied lambda calculi. *Logical Methods in Computer Science*, 1(2):1–14, 2005.

[5] U. Berger and P. Oliva. Modified bar recursion and classical dependent choice. *Lecture Notes in Logic*, 20:89–107, 2005.

[6] U. Berger and P. Oliva. Modified bar recursion. *Mathematical Structures in Computer Science*, 16(2):163–183, 2006.

[7] M. Bezem. Strongly majorizable functionals of finite type: A model for bar recursion containing discontinuous functionals. *Journal of Symbolic Logic*, 50:652–660, 1985.

[8] M. Bezem. Equivalence of bar recursors in the theory of functionals of finite type. *Archive for Mathematical Logic*, 27:149–160, 1988.

[9] T. Coquand. Constructive topology and combinatorics. In *Constructivity in Computer Science*, volume 613 of *LNCS*, pages 159–164, 1991.

[10] Y. L. Ershov. Model C of partial continuous functionals. In *Logic Colloquium*, pages 455–467. North Holland, Amsterdam, 1977.

[11] M. Escardó. Synthetic topology of data types and classical spaces. *ENTCS*, 87:21–156, 2004.

[12] M. Escardó and P. Oliva. Computational interpretation of analysis via products of selection functions. In *Proceedings of CiE 2010*, volume 6158 of *LNCS*, pages 141–150. 2010.

[13] M. Escardó and P. Oliva. Selection functions, bar recursion and backward induction. *Mathematical Structures in Computer Science*, 20(2):127–168, 2010.

[14] M. Escardó and P. Oliva. Sequential games and optimal strategies. *Royal Society Proceedings A*, 467:1519–1545, 2011.

[15] M. Escardó and P. Oliva. Computing Nash equilibria of unbounded games. In *Proceedings of the Turing Centenary Conference, Manchester*, volume 10 of *EPiC Series*, pages 53–65, 2012.

[16] M. Escardó and P. Oliva. The Peirce translation. *Annals of Pure and Applied Logic*, 163(6):681–692, 2012.

[17] M. Escardó and P. Oliva. Bar recursion and products of selection functions. In press, preprint available from author's webpage at http://www.cs.bham.ac.uk/∼mhe/papers/index.html, 2013.

[18] R. Gandy and M. Hyland. Computable and recursively countable functionals of higher type. In R. Gandy and M. Hyland, editors, *Logic Colloquium 1976*, pages 407–438. North-Holland, Amsterdam, 1977.

[19] S. C. Kleene. Countable functionals. In A. Heyting, editor, *Constructivity in Mathematics*, pages 81–100. North-Holland, Amsterdam, 1959.

[20] U. Kohlenbach. *Theory of Majorizable and Continuous Functionals and their Use for the Extraction of Bounds from Non-Constructive Proofs: Effective Moduli of Uniqueness for Best Approximations from Ineffective Proofs of Uniqueness (German)*. PhD thesis, Frankfurt, 1990.

[21] U. Kohlenbach. *Applied Proof Theory: Proof Interpretations and their Use in Mathematics*. Monographs in Mathematics. Springer, 2008.

[22] G. Kreisel. Interpretation of analysis by means of functionals of finite type. In A. Heyting, editor, *Constructivity in Mathematics*, pages 101–128. North-Holland, Amsterdam, 1959.

[23] C. St. J. A. Nash-Williams. On well-quasi-ordering finite trees. *Proceedings of the Cambridge Philosophical Society*, 59:833–835, 1963.

[24] P. Oliva and T. Powell. On Spector's bar recursion. *Mathematical Logic Quarterly*, 58:356–365, 2012.

[25] T. Powell. *On Bar Recursive Interpretations of Analysis*. PhD thesis, Queen Mary University of London, 2013.

[26] J.-C. Raoult. Proving open properties by induction. *Information Processing Letters*, 29:19–23, 1988.

[27] M. Seisenberger. *On the Constructive Content of Proofs*. PhD thesis, Ludwig Maximilians Universität München, 2003.

[28] M. Seisenberger. Programs from proofs using classical dependent choice. *Annals of Pure and Applied Logic*, 153(1-3):97–110, 2008.

[29] 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.

[30] A. S. Troelstra. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*, volume 344 of *Lecture Notes in Mathematics*. Springer, Berlin, 1973.