

Parametrised bar recursion: A unifying framework for realizability interpretations of classical dependent choice

Thomas Powell
Institute of Computer Science, University of Innsbruck*

March 13, 2015

Abstract

During the last twenty years or so a wide range of realizability interpretations of classical analysis have been developed. In many cases, these are achieved by extending the base interpreting system of primitive recursive functionals with some form of bar recursion, which realizes the negative translation of either countable or countable dependent choice. In this work we present the many variants of bar recursion used in this context as instantiations of a general, parametrised recursor, and give a uniform proof that under certain conditions this recursor realizes a corresponding family of parametrised dependent choice principles. From this proof, the soundness of most of the existing bar recursive realizability interpretations of choice, including those based on the Berardi-Bezem-Coquand functional, modified realizability and the more recent products of selection functions of Escardó and Oliva, follows as a simple corollary. We achieve not only a uniform framework in which familiar realizability interpretations of choice can be compared, but show that these represent just simple instances of a large family of potential interpretations of dependent choice principles.

Keywords: Dependent choice, Modified realizability, Open induction, Bar recursion, Continuous functionals.

1 Introduction

One of the central problems of mathematical logic and computer science is to understand the constructive meaning of non-constructive proofs. From the early 20th century onwards a rich variety of constructive interpretations of classical logic have been developed, together with techniques for extracting computational content from proofs. The majority of these techniques initially deal with proofs in formal systems of classical predicate logic or weak subsystems of mathematics such as Peano arithmetic. In order to interpret stronger non-constructive proofs from subsystems of mathematical analysis which contain principles such as the axiom of dependent choice,

$$\text{DC} : \forall n, x^\rho \exists y^\rho A_n(x, y) \rightarrow \exists f^{\mathbb{N} \rightarrow \rho} \forall n A_n(f(n), f(n+1)),$$

these techniques must typically be adapted and extended - a process which tends to be highly non-trivial.

In this paper we focus on just one method of giving a computational interpretation to proofs: namely classical modified realizability. A long established way of extending this technique to deal with choice principles is to introduce some kind of bar recursion to the usual interpreting system of primitive recursive functionals. Bar recursive interpretations of choice principles are among the most widely known and studied methods for giving a computational interpretation to classical analysis, and numerous instances of this method can be found in the literature. However, both the particular variant of bar recursion used and the form of choice that it realizes differs from case to case, and it is not well-understood how these variants compare as realizers. This is compounded by the fact that each variant was typically devised in a distinct setting, and both their existence and correctness often established using a slightly different method of proof, obscuring the inherent similarities between them all.

***Email:** Thomas.Powell@uibk.ac.at **Tel:** +43 512 507 53293

The main contribution of this paper is to construct a general bar recursive term which contains several free parameters. We show that whenever these parameters obey certain conditions, the term can be used to realize the negative translation of a corresponding variant of countable dependent choice, and that moreover essentially all of the known variants of bar recursion used to extend modified realizability to classical analysis appear as simple instantiations of the parameters.

Our motivation for this is twofold. Firstly, we obtain a clear, unifying framework in which the differences between existing variants of bar recursion and their correctness proofs are essentially reduced to simple structures on the natural numbers, and thus their behaviour as realizers is more easily compared. Secondly, in doing this we greatly generalise the interpretation of classical analysis by giving not one but a whole class of realizers of choice principles, which can be freely chosen between to suit the problem at hand. Having a such a wide range of options in front of us means that in concrete instances of program extraction we are given the potential to tailor our realizer to the situation at hand and extract computational content which is more efficient and semantically meaningful.

1.1 Realizability interpretations of classical analysis: A brief overview

This work is designed to be as self-contained as possible, and in particular no prior knowledge of bar recursion is assumed. However, because we are partly motivated by the desire to unify existing interpretations, we provide here a very short summary of some of the best known variants of bar recursion that we aim to bring together. We simply state without explanation their defining equations: the purpose here is not to treat in any detail the exact meaning of these objects, but to allow the reader to see certain key features of the recursion that will be dealt with in a more general setting later.

The idea of interpreting countable choice with bar recursion was first established not for realizability interpretations, but for Gödel's slightly more intricate Dialectica interpretation, in a fundamental paper of Spector in 1962 [27]. Here, bar recursion was given as the schema

$$\text{SBR}(Y, G, H, s^{X^*}) = \begin{cases} G(s) & \text{if } Y(\widehat{s}) < |s| \\ H(s, \lambda x . \text{SBR}(Y, G, H, s * x)) & \text{otherwise.} \end{cases} \quad (1)$$

The key details here are that $s : X^*$ is a finite sequence, $s * x$ denotes the extension of s with the object x , \widehat{s} its canonical extension as an infinite sequence, and the output type of Y is a natural number. Spector's bar recursion is therefore a form of 'backward recursion' in which recursive calls are made on extensions $s * x$ of the argument s . The parameter Y is responsible for terminating the recursion, based on the assumption that the underlying tree barred by sequences satisfying $Y(\widehat{t}) < |t|$ is well-founded - a fact that is true in all continuous models and also in non-continuous structures such as the strongly majorizable functionals [9]. What this means is that for any sequence $s * x_1, s * \langle x_1, x_2 \rangle, \dots$ of recursive calls made in the computation of $\text{SBR}(Y, G, H, s)$, there is always some N such that $Y(\widehat{s * \langle x_1, \dots, x_N \rangle}) < |s| + N$, and thus $\text{SBR}(Y, G, H, s)$ is a well-defined value.

An early adaptation of this idea to realizability was given by Berardi et al. [3], in which an interpretation for dependent choice broadly similar but somewhat simpler than Spector's variant of bar recursion was established. This realizer was later reformulated in the more standard framework of modified realizability in [6], where it was given its now familiar name *modified bar recursion* and was defined as

$$\text{MBR}(Y, H, s) =_{\mathbb{N}} Y(s \oplus \lambda n^{\mathbb{N}} . H(s, \lambda x . \text{MBR}(Y, H, s * x))). \quad (2)$$

Here s is again a finite sequence, $s \oplus \alpha$ denotes the infinite sequence α overwritten by s , and the outcome type of Y is a natural number. This restriction on the type of Y ensures that in continuous models Y only looks at a finite amount of information from its input sequence. In other words, given a sequence of recursive calls $s * x_1, s * \langle x_1, x_2 \rangle, \dots$ in the computation of $\text{MBR}(Y, H, s)$, there is some point N such that the value of $q(s * \langle x_1, \dots, x_N \rangle \oplus \dots)$ is determined based only on $s * \langle x_1, \dots, x_N \rangle$, and so no further recursive calls are necessary. Thus, like Spector's variant of bar recursion (1), modified bar recursion is carried out over some underlying well-founded tree, although unlike Spector's bar recursion, this tree is not computable but it *implicitly* well-founded by some kind of continuity argument.¹

¹Indeed, a fact closely related to this observation, proved in [7], is that (2) is not S1-S9 computable in the total continuous functionals and is therefore strictly stronger than (1).

In addition to what later became modified bar recursion, [3] contains a striking realizer for (the negative translation of) countable choice, which took from Spector only the basic idea of backward recursion, replacing the sequential bar recursive calls $s \mapsto s * x$ of (1) and (2) with a symmetric updating $u \mapsto u_n^x$ of finite partial functions. This realizer was again put into a standard realizability framework in [4, 5], and will be referred to here as the Berardi-Bezem-Coquand, or BBC-functional. Its defining equation (following [4]) is given by

$$\text{BBC}(Y, H, u) =_{\mathbb{N}} Y(u \oplus \lambda n. H(n, \lambda x . \text{BBC}(u_n^x))). \quad (3)$$

Here, u is a finite partial function and u_n^x is the domain-theoretic extension of u with value x at input n . As with modified bar recursion, BBC terminates by continuity of Y , although proving this is somewhat technical and as shown in [5] is most elegantly done with Zorn’s lemma in the form of open induction. It has been argued that (3) provides a computational interpretation of countable choice that is superior to standard bar recursion in that it is ‘demand driven’, in the sense that to compute an approximation for the n th point in a choice sequence it is not automatically necessary to compute approximations for $1, \dots, n - 1$ first [3].

More recently, a family of new variants of bar recursion known as *products of selection functions* were developed and explored by Escardó and Oliva, beginning in [13]. One of these, the so-called *implicit product of selection functions*, was shown in [15] to realize not only the negative translation of countable dependent choice but more generally a dependent version of the so called J -shift arising from the Pierce translation. The product of selection functions is distinguished by the fact that it incorporates course-of-values recursion into its bar recursive calls. It can be formulated as

$$\text{IPS}(Y, H, s) =_{\mathbb{N}} Y(s \oplus \lambda n. H(t_n, \lambda x . \text{IPS}(Y, H, t_n * x))) \quad (4)$$

where t_n is the sequence of length n primitive recursively defined by

$$(t_n)_i = s_i \text{ if } i < |s|, \text{ else } H(t_i, \lambda x . \text{IPS}(Y, H, t_i * x)) \text{ for } i < n.$$

In addition to solving the modified realizability interpretation of choice, this form of bar recursion has deep links to game theory as a functional that computes optimal strategies in a class of higher-type, continuously well-founded sequential games [14]. This provides a highly illuminating bridge between the computational content of the axiom of choice and the world of game theory, and is exploited in e.g. [23] to give a game-theoretic constructive interpretation of Ramsey’s theorem.

Of course, this list is by no means exhaustive, and several further variants of (2)-(4) have been devised for realizing choice principles. For example, a realizer of the refined A -translation of a sequential variant of dependent choice is given in [26], and is used in [25] to extract a realizer for Higman’s lemma. Similarly, forms of bar recursion closely related to (2) have been used for realizing dependent choice in a range of settings, including Parigot’s $\lambda\mu$ -calculus [10] and in the context of realizability toposes [28].

Nevertheless, all of these recursors share two important features in common:

1. they take as input some partial object t , and make recursive calls over extensions of this object;
2. they terminate because the value of $Y(\alpha)$ only depends on a finite portion of the input α .

In fact, one could go as far as to say that (2)-(4) along with their many other variants are essentially the same realizer, with the single exception that recursive calls are made using slightly different patterns. In this paper we make this idea precise, and show that the basic recipe used to form the realizers mentioned above can be used to construct, in a completely uniform way, an infinite class of bar recursive functionals, each one of which can be used to interpret a particular set of variants of the axiom of dependent choice.

Note that we restrict our attention to interpretations of analysis based on modified realizability, and in particular bar recursors of the form (1) which arise from the Dialectica interpretation do not fit into our framework. Nevertheless, we believe that the basic ideas behind this work could be readily lifted to the Dialectica interpretation (see for example the author’s recent article with P. Oliva [22] on a symmetric form of (1) which updates partial functions similarly to the BBC-functional), and may also be helpful in linking proof interpretations to more direct computational interpretations of analysis, such as the learning-based realizabilities of e.g. [1, 2].

1.2 Outline of the paper

After setting up our basic formal systems in Section 2, we define in Section 3 a general principle of backward induction which will be used throughout the paper in order to prove the correctness of our realizers. Backward induction replaces principles such as bar induction or dependent choice which are more typically used on the meta-level to prove correctness of realizers, and is preferred here due to its far greater flexibility. We also define an analogous schema of backward recursion which will be used to construct our parametrised realizers. This section is strongly influenced by Berger [5], and our formulation of backward induction is similar to but slightly more general than his update induction.

Sections 4 and 5 form the core of the paper, in which we define our parametrised realizer and use it to realize dependent choice. In Section 4 we restrict ourselves to the simpler double negation shift: While this section's main result, Theorem 4.6, is eventually subsumed by the later Theorem 5.3, it is instructive to first focus on the double negation shift so that the main ideas of Section 5 can be appreciated. In the sequel we introduce a family of dependent choice principles parametrised by a well-founded ordering \triangleleft on the natural numbers, and show that the negative translation of each of these principles can be realized by our parametrised bar recursor under certain conditions. Theorem 5.3 gives essentially all the results listed in Section 1.1 as a Corollary, and vastly generalises all of them.

We conclude on a more informal level, and in Section 6 discuss a potential semantic interpretation of our parametrised realizer, inspired by the game-theoretic semantic reading of the Berardi-Bezem-Coquand functional in [3].

2 Preliminaries

Throughout this paper we work in variant of extensional Heyting/Peano arithmetic in all finite types, which will be essentially the $\text{E-HA}^\omega/\text{E-PA}^\omega$ as defined in e.g. [17, 29], but based on a slightly richer type system.

2.1 The finite types

For us, the finite types consist of base types \mathbb{N} and \mathbb{B} for natural numbers and booleans, and are build from the formation of function types $\rho \rightarrow \tau$, product types $\rho \times \tau$, and finite sequence types ρ^* . We sometimes use the abbreviation τ^ρ for $\rho \rightarrow \tau$.

A *discrete* type is any type τ which can be encoded in \mathbb{N} , and so in all standard models \mathbb{B} , \mathbb{N} , $\mathbb{B} \times \mathbb{N}$, \mathbb{N}^* etc. are discrete types but $\mathbb{N} \rightarrow \mathbb{N}$ is not.

2.2 The theory E-HA^ω

Terms of E-HA^ω are typed lambda terms formed by application and abstraction, and include variables for each type, the usual constructors and deconstructors for product and sequence types, the arithmetic constants $0: \mathbb{N}$ and $s: \mathbb{N} \rightarrow \mathbb{N}$, and finally recursors $R_\rho: \rho \rightarrow (\mathbb{N} \rightarrow \rho \rightarrow \rho) \rightarrow \mathbb{N} \rightarrow \rho$ of each type.

Equations in E-HA^ω are formed using basic symbols $=_{\mathbb{B}}$ and $=_{\mathbb{N}}$ for equality of type \mathbb{B} and \mathbb{N} , while formulas of E-HA^ω are built from $=_{\mathbb{N}}$ and $=_{\mathbb{B}}$ together with the usual logical connectives and quantifiers for all types. Equality at arbitrary types is defined inductively, so for example $f =_{\rho \rightarrow \tau} g$ abbreviates $\forall x^\rho (fx =_\tau gx)$.

The axioms and rules of E-HA^ω are the standard axioms and rules of classical logic in all finite types, along with those of the typed lambda calculus, defining equations for all the constants, induction for arbitrary formulas, and finally full extensionality:

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

In what follows we often consider extensions of E-HA^ω with new recursively defined functionals F , in which case by $\text{E-HA}^\omega + F$ we mean the extension of E-HA^ω with new constants F and their associated defining axioms.

We use, throughout, the following notation and abbreviations.

- 0_ρ is the inductively defined 0 term of type ρ .

- π_i and \langle , \rangle denote the projection and pairing operations. For a sequence $\alpha: (\rho_0 \times \rho_1)^{\mathbb{N}}$ we sometimes write $\alpha_i: \rho^{\mathbb{N}}$ for $\lambda n. \pi_i(\alpha(n))$.
- For a finite sequence $s: \rho^*$, $|s|$ denotes the length of s , while $s * t := \langle s_0, \dots, s_{m-1}, t_0, \dots, t_{n-1} \rangle$ denotes the concatenation s and t . We also use $s * x$ to denote $s * \langle x \rangle$, and $s * \alpha$ for the concatenation of s with the infinite sequence α .
- $s \oplus \alpha: \rho^{\mathbb{N}}$ denotes the overwriting of α with the finite sequence s i.e.

$$(s \oplus \alpha)(n) := \begin{cases} s_n & \text{if } n < |s| \\ \alpha(n) & \text{if } n \geq |s| \end{cases}.$$

- $[\alpha](n) := \langle \alpha(0), \dots, \alpha(n-1) \rangle$ denotes the finite initial segment of length n of $\alpha: \rho^{\mathbb{N}}$, while \widehat{s} denotes the canonical extension $s \oplus 0_{\rho^{\mathbb{N}}}$ of the finite sequence $s: \rho^*$.
- For a decidable predicate $P(x)$, the term ' y^ρ if $P(x)$ ' of type ρ is shorthand for

$$\begin{cases} y & \text{if } P(x) \\ 0_\rho & \text{otherwise} \end{cases}.$$

- For a decidable predicate $P(n)$ on \mathbb{N} , the term $\mu i \leq n. P(i): \mathbb{N}$ is the least $i \leq n$ satisfying $P(i)$, and just n if no such i exists.

In addition to all this, throughout the article we work with a type of partial sequences with potentially infinite domain. These can be encoded, for example, by total sequences of type $\mathbb{N} \rightarrow \mathbb{B} \times \rho$, where defined values are represented as $(1, x)$, and undefined values as $(0, 0_\rho)$. Accordingly, for an object u of this type we say that n is in the domain of u , or $n \in \text{dom}(u)$, if $\pi_0 u(n) = 1$, and $n \notin \text{dom}(u)$ otherwise. Membership of $\text{dom}(u)$ is a decidable property.

We imagine $\mathbb{B} \times \rho$ as simulating a type $\bar{\rho} \equiv \rho + 1$, and we write $u(n) =_{\bar{\rho}} x$ instead of $u(n) =_{\mathbb{B} \times \rho} (1, x)$, and $u(n) =_{\bar{\rho}} \perp$ whenever n is not in the domain of u . Similarly, $u =_{\bar{\rho}} v$ if for all n we have either $\pi_0 u(n) = \pi_0 v(n) = 0$ or $\pi_0 u(n), \pi_0 v(n) = 1$ and $\pi_1 u(n) = \pi_1 v(n)$.

- We use the symbol \emptyset to denote the partial function with empty domain.
- We extend the overwrite operation \oplus given for finite sequences above to partial functions by defining $u \oplus v: \bar{\rho}^{\mathbb{N}}$ by

$$(u \oplus v)(n) = \begin{cases} u(n) & \text{if } n \in \text{dom}(u) \\ v(n) & \text{if } n \notin \text{dom}(u). \end{cases}$$

We define $u \oplus \alpha: \rho^{\mathbb{N}}$ for total sequence $\alpha^{\mathbb{N}}$ analogously. It will always be clear from the context which types the operator \oplus takes as input.

- We isolate as a special case the addition of a single value to u : For $n^{\mathbb{N}}$ and x^ρ we define $u_n^x = u \oplus (n, x)$ where (n, x) is the partial function taking defined value x at point n , and undefined elsewhere. When $n \notin \text{dom}(u)$ we say that u_n^x is an *update* of u .
- Finally, we write $u \sqsupseteq v$ whenever $\forall i \in \text{dom}(v)(u(i) = v(i))$. Note that we have $u \oplus v \sqsupseteq u$ for any v , and $u \oplus v \sqsubset u$ whenever there is some $n \notin \text{dom}(u)$ such that $n \in \text{dom}(v)$. We also have $u_n^x \sqsubset u$ when $n \notin \text{dom}(u)$.

For us, the system E-HA^ω together with its classical variant E-PA^ω acts as a standard lambda calculus equipped with a robust meta-theory for reasoning about terms. However, the exact details of our formal system are not particularly important as everything which follows can be easily lifted to alternative settings. For example, a slightly different approach would be to work a weaker, quantifier-free term calculus T and to do all the reasoning in an unspecified meta-theory, as in [3]. Alternatively we could work in a theory of partial continuous functionals as in [4], taking our base type to represent the flat domain \mathbb{N}_\perp .

2.3 Models of E-HA^ω

In order to prove both the existence and correctness of bar recursive realizers, it is typically necessary to work in a constructive interpretation $T \equiv (T_\rho)$ of E-HA^ω which satisfies some form of the following two properties:

1. (T_ρ) contains *arbitrary* choice sequences, in other words $T_{\mathbb{N} \rightarrow \rho}$ contains all sequences $\mathbb{N} \rightarrow T_\rho$ and so in particular (T_ρ) validates dependent choice;
2. Whenever τ is a discrete type, functionals of type $F: \rho^{\mathbb{N}} \rightarrow \tau$ satisfy the following continuity principle:

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

Both of these principles are satisfied automatically by the Kleene-Kreisel continuous functionals \mathcal{C}^ω [16, 18], whereas for term models such as the theory \mathcal{P} of [3], (1) is obtained by adding infinite choice sequences explicitly. Here we do not choose any particular interpretation of E-HA^ω, rather we simply add principles such as Cont and dependent choice to our meta-theory whenever they are required.

3 Backward induction and recursion

We now develop some of the crucial background theory that will be required in order to prove our main results. In particular, we formulate a general principle of backward induction and define an associated backward recursor, both of which will be used to construct and verify the correctness of our parametrised realizer. In simple terms, backward induction is induction over domain-theoretic extensions of partial sequences, and should be seen as a generalisation of bar induction. Analogously, backward recursion is a generalisation of the implicit forms of bar recursion used to give realizability interpretations to countable choice. This section is largely inspired by [5] in that we formulate backward induction as an instance of the still more general principle of open induction.

3.1 Open induction

Open induction, first considered by Raoult in [24], is an extension of well-founded (or Noetherian) induction to chain-complete partial orders. Recall that a partial order (X, \leq) is chain-complete if every non-empty chain γ in X has a least upper bound $\bigvee \gamma$. A predicate B on X is *open* if it satisfies the property

$$B(\bigvee \gamma) \rightarrow \exists x \in \gamma B(x)$$

for every non-empty chain γ in X , and the principle of open induction over X is given by

$$\text{OI}_{(X, \leq)} : \forall x (\forall y > x B(y) \rightarrow B(x)) \rightarrow \forall x B(x)$$

where B ranges over open predicates. Note that open induction implies well-founded induction since whenever $>$ is well-founded (X, \leq) is trivially chain-complete and all predicates are automatically open. However, in general $>$ need not be well-founded, in which case openness becomes a non-trivial property.

Theorem 3.1 (Raoult [24]). *Any chain-complete partial order satisfies open induction.*

Proof. This is a direct consequence of Zorn's lemma. Suppose that the open predicate B satisfies the premise of open induction, which is classically equivalent to

$$\forall x (\neg B(x) \rightarrow \exists y > x \neg B(y)), \tag{5}$$

and suppose for contradiction that the set

$$S = \{x \in X \mid \neg B(x)\}$$

is non-empty. We show that every chain in S has an upper bound in S . For the empty chain this is trivial since S contains at least one element. On the other hand, if γ is non-empty, then it has an upper bound $\bigvee \gamma$ in X by completeness, and moreover $\bigvee \gamma \in S$ since $\forall x \in \gamma \neg B(x) \rightarrow \neg B(\bigvee \gamma)$ by openness of B . Therefore by Zorn's lemma S contains a maximal element, which contradicts (5). Thus S must be empty. \square

3.2 Backward induction

We are now ready to define backward induction, which we simply take to be open induction over the partial order $(\mathbb{N} \rightarrow \bar{\rho}, \sqsubseteq)$, where \sqsubseteq is the extension relation on partial sequences defined in Section 2.2. In other words, backward induction is the schema

$$\text{Bkl} : \forall u^{\mathbb{N} \rightarrow \bar{\rho}} (\forall v \sqsubset u B(v) \rightarrow B(u)) \rightarrow \forall u B(u)$$

where B ranges over open predicates. This formulation of backward induction makes sense and is valid in any interpretation T of E-PA^ω that admits arbitrary sequences, since for any chain γ in $T_{\mathbb{N} \rightarrow \bar{\rho}}$,

$$(\bigvee \gamma)(n) := \begin{cases} u(n) & \text{if } n \in \text{dom}(u) \text{ for some } u \in \gamma \\ \perp & \text{otherwise.} \end{cases}$$

is a perfectly well-defined object of $\mathbb{N} \rightarrow T_{\bar{\rho}} \equiv T_{\mathbb{N} \rightarrow \bar{\rho}}$ and is the least upper bound of γ with respect to \sqsubseteq . It will be convenient to isolate the following syntactic notion of an open formula, which will be sufficient for everything that follows.

Proposition 3.2. *Suppose that the formula $B(u)$ on partial sequences is of the form*

$$B(u) := \forall n [n \in \text{dom}(u) \rightarrow A(n, u)] \rightarrow \exists n P([u](n))$$

where P is an arbitrary predicate on $\bar{\rho}^*$, and A is monotone in the following sense

$$n \in \text{dom}(u) \wedge u \sqsubseteq v \wedge A(n, u) \rightarrow A(n, v). \quad (6)$$

Then $B(u)$ is open with respect to \sqsubseteq .

Proof. Given some chain γ let $v = \bigvee \gamma$ and assume that $B(v)$ holds. We must show that $B(u)$ holds for some $u \in \gamma$. If $B(v)$ holds, then by classical logic we either have $P([v](n))$ for some n , in which case $B(u)$ holds for e.g. the least $u \in \gamma$ satisfying $[u](n) = [v](n)$, or we have

$$n \in \text{dom}(v) \wedge \neg A(n, v)$$

for some n . In this latter case pick the least $u \in \gamma$ satisfying $n \in \text{dom}(u)$. Then since $u \sqsubseteq v$, we have $A(n, u) \rightarrow A(n, v)$ by the monotonicity condition, which is a contradiction. Therefore $\neg A(n, u)$ and $B(u)$ holds. \square

Remark 3.3. Note that Berger [5] generally studies formulas which are open with respect to the *lexicographic ordering* on infinite sequences, which are not necessarily open with respect to \sqsubseteq (which is strictly contained in a lexicographic ordering). However, any formula equivalent to one of the form $B(u) := \exists n P([u](n))$ is open in both senses.

3.3 Some additional remarks on backward induction

We take the opportunity to explore backward induction in more detail and relate it to other well-known principles in logic, including the minimal bad sequence argument and bar induction. This section is not strictly necessary for the remainder of the paper, so if the reader prefers they can proceed directly to the definition of backward recursion given in Section 3.4.

We first point out that, analogously to [5], backward induction does not require the full strength of Zorn's lemma, and is provable from just dependent choice, using a version of the minimal-bad-sequence argument due to Nash-Williams [19].

Proposition 3.4. *The principle of backward induction is provable in $\text{E-PA}^\omega + \text{DC}$.*

Proof. Take some open formula $B(u)$, which for simplicity we assume is open in the sense of Proposition 3.2. Suppose for contradiction that we have $\forall u (\neg B(u) \rightarrow \exists v \sqsubset u \neg B(v))$ but there exists some partial sequence u_0 such that $\neg B(u_0)$. Using dependent choice construct the sequence (u_n) as follows: Supposing that we have already constructed $\langle u_0, \dots, u_n \rangle$ for $n \geq 0$, define

- (i) $u_{n+1} := w$ if $n \notin \text{dom}(u_n)$ and w is such that the following four properties are satisfied: $\neg B(w)$;
 $w \sqsupset u_n$; $[w](n) = [u_n](n)$ and $n \in \text{dom}(w)$,
- (ii) $u_{n+1} := u_n$ if either $n \in \text{dom}(u_n)$ or $n \notin \text{dom}(u_n)$ and no w in the sense of (i) exists.

First, it is clear by a simple induction that for all n we have

- (a) $\neg B(u_n)$,
- (b) $[u_n](n) = [u_{n+1}](n)$, and
- (c) $u_n \sqsubseteq u_{n+1}$.

Define $\tilde{u} := \lambda n. u_{n+1}(n)$. Then it follows that for all n we have

- (d) $[\tilde{u}](n) = [u_n](n)$ and
- (e) $u_n \sqsubseteq \tilde{u}$.

The first of these is done by a simple induction using (b). For the latter, take $i \in \text{dom}(u_n)$. Then either $i < n$ in which it is clear by (e) that $\tilde{u}(i) = u_n(i)$, or $i \geq n$ and we obtain $\tilde{u}(i) = u_{i+1}(i) = u_n(i)$ by $u_n \sqsubseteq \dots \sqsubseteq u_{i+1}$.

Now we prove that $\neg B(\tilde{u})$, which is classically equivalent to

$$\forall i([i \in \text{dom}(u) \rightarrow A(i, u)] \wedge \neg P([u](i))).$$

Taking some arbitrary n , and setting $u = u_{n+1}$ and $i = n$ we get, by $\neg B(u_{n+1})$ (true by (a)),

$$(*) [n \in \text{dom}(u_{n+1}) \rightarrow A(n, u_{n+1})] \wedge \neg P([u_{n+1}](n)).$$

But $\neg P([u_{n+1}](n)) \rightarrow \neg P([\tilde{u}](n))$ by (b) and (d), and furthermore $n \in \text{dom}(\tilde{u})$ is equivalent to $n \in \text{dom}(u_{n+1})$ and hence by (*) implies $A(n, u_{n+1})$, so using monotonicity of A and the fact that $u_{n+1} \sqsubseteq \tilde{u}$ we have $A(n, \tilde{u})$ and have therefore established

$$n \in \text{dom}(\tilde{u}) \rightarrow A(n, \tilde{u}).$$

Taken together and bearing in mind that n is arbitrary, this implies $\neg B(\tilde{u})$. But now we know by the backward induction hypothesis that there exists some $v \sqsupset \tilde{u}$ such that $\neg B(v)$ holds.

We can show that this contradicts the construction of (u_n) , and therefore there cannot exist any initial sequence u_0 satisfying $\neg B(u_0)$, and we're done. Let m be the least point such that $m \in \text{dom}(v)$ but $m \notin \text{dom}(\tilde{u})$. Then firstly by (e) we have $u_m \sqsubseteq \tilde{u} \sqsubset v$ and thus $u_m \sqsubset v$, and secondly $[v](m) = [\tilde{u}](m) = [u_m](m)$, the first equality by minimality of m and the second by (d), and therefore v satisfies the required properties of w in (i) at point m . In addition we know that $m \notin \text{dom}(u_m)$ else we'd have $m \in \text{dom}(\tilde{u})$ by (e), and therefore u_{m+1} must be constructed using (i) and thus $m \in \text{dom}(u_{m+1}) \subseteq \text{dom}(\tilde{u})$, contradicting the assumption that $m \notin \text{dom}(\tilde{u})$. \square

As we will see in Sections 4 and 5, one of the key ideas in this paper is construct forms of recursion based on restricted, or relativised variants of backward recursion which take as input partial functions that are downward closed with respect to some relation on \mathbb{N} . These will be closely related to the following form of relativised backward induction.

Proposition 3.5. *Let \triangleleft be some decidable relation on \mathbb{N} , and define the predicate $u \in D_{\triangleleft}$ by*

$$u \in D_{\triangleleft} :\equiv \forall n \in \text{dom}(u)[\forall i \triangleleft n(i \in \text{dom}(u))].$$

Equivalently, we say that $\text{dom}(u)$ is \triangleleft -closed. Then for any relation \square' on $\bar{\rho}^{\mathbb{N}}$ such that $u \square' v \rightarrow u \sqsubset v$, the following principle of relativised backward induction is provable from Bkl:

$$\forall u \in D_{\triangleleft}(\forall v \sqsupset' u[v \in D_{\triangleleft} \rightarrow B(v)] \rightarrow B(u)) \rightarrow \forall u \in D_{\triangleleft} B(u).$$

Proof. First note that $u \in D_{\triangleleft}$ is of the form $\forall n \in \text{dom}(u) D_0(n, u)$ with $D_0(n, u)$ monotone in the sense of (6), therefore the predicate $B'(u) := u \in D_{\triangleleft} \rightarrow B(u)$ is open for any open $B(u)$. Thus we obtain

$$\forall u (\forall v \sqsupset' u B'(v) \rightarrow B'(u)) \rightarrow \forall u (\forall v \sqsupset u B'(v) \rightarrow B'(u)) \rightarrow \forall u B'(u)$$

the first implication following from the inclusion $\sqsupset' \subseteq \sqsupset$ and the second from normal backward induction applied to $B'(u)$. Rearranging this gives us relativised backward induction. \square

We can now instantiate \triangleleft and \sqsupset' to obtain certain well-known instances of backward induction.

Example 3.6 (Update induction). If we define $u \sqsupset' v$ iff v is an update of u , and let \triangleleft just be the empty relation, then relativised backward induction just becomes update induction in the sense of [5].

Example 3.7 (Bar induction). Now consider the case $\triangleleft = <$. By classical logic, if $u \in D_{<}$ then either $u = \alpha$ for some total sequence α or $u = \widehat{s}$ where s is some finite sequence and \widehat{s} its embedding as a partial function. Therefore relativised backward induction is equivalent in $\mathbf{E-PA}^\omega$ to

$$\forall \alpha B(\alpha) \wedge \forall s (\forall v \sqsupset' \widehat{s} [v \in D_{<} \rightarrow B(v)] \rightarrow B(\widehat{s})) \rightarrow \forall s B(\widehat{s}) \quad (7)$$

In addition, if we define $u \sqsupset' v$ iff v is an update of u , then (7) becomes equivalent to

$$\forall \alpha B(\alpha) \wedge \forall s (\forall x B(\widehat{s * x}) \rightarrow B(\widehat{s})) \rightarrow \forall s B(\widehat{s}) \quad (8)$$

which is just a variant of bar induction. It is not too difficult to show that this is equivalent over $\mathbf{E-PA}^\omega$ to the more standard formulations of bar induction found in e.g. [29].

3.4 Backward recursion

The purpose of introducing backward induction was to give us a way to reason about backward recursion, which we define and discuss in this section. In the same way that backward induction is a special case of open induction, backward recursion is closely related (and in fact definable from) open recursion as defined in [5].

To begin with, in order to motivate what follows let us consider as a comparison the entirely standard concept of well-founded recursion over some decidable well-founded relation \prec on ρ . Assuming we are working in a structure such as PCF or the Scott continuous functionals, we can define a well-founded recursor \mathbf{wR}_\prec as the fixpoint of the following recursive equation

$$\mathbf{wR}_\prec^\psi(x) =_\sigma \psi_x(\lambda y . \mathbf{wR}_\prec^g(y) \text{ if } y \prec x),$$

and prove that the recursor defines a *total* functional for any outcome type σ using well-founded induction over \prec :

$$\mathbf{wl} : \forall x (\forall y \prec x A(y) \rightarrow A(x)) \rightarrow \forall x A(x).$$

We want to define a backward recursor in a similar way - although we have two problems: firstly the relation $u \sqsupset v$ is not decidable, and secondly backward induction is only valid for open formulas. We avoid these issues by defining backward recursion to be the fixpoint of the following recursive equation

$$\mathbf{BkR}_{\rho, \tau}^\psi(u) =_\tau \psi_u(\lambda n, v . \mathbf{BkR}^\psi(u \oplus v) \text{ if } n \in \text{dom}(v) \setminus \text{dom}(u)),$$

where τ is restricted to being a discrete type, while $n \in \text{dom}(v) \setminus \text{dom}(u)$ denotes the decidable predicate $n \in \text{dom}(v) \wedge n \notin \text{dom}(u)$. Observe that any $w \sqsupset u$ is of the form $w = u \oplus v$ for some v which is defined at at least one point $n \notin \text{dom}(u)$, and so \mathbf{BkR} makes recursive calls over all $w \sqsupset u$, although crucially it must always have access to a point $n \in \text{dom}(w)$ such that $n \notin \text{dom}(u)$. The necessity of the restriction on τ is to ensure that totality of $\mathbf{BkR}^\psi(u)$ is an open property on total input u (here we are referring to total elements of the *model* of partial continuous functionals as opposed to the type $\bar{\rho}$). Indeed for discrete τ and total u we have

$$\mathbf{BkR}^\psi(u) \text{ is total} \leftrightarrow \exists n \forall w (\mathbf{BkR}^\psi([u](n) \oplus w) \text{ is total})$$

assuming sequential continuity \mathbf{Cont} for functionals with total output. Therefore totality of \mathbf{BkR} is provable using backward induction. Note that alternatively, a direct proof via Zorn's lemma that \mathbf{BkR}

exists as a total element of the Scott partial continuous functionals can be carried out using the same manner as the proof of totality of the Berardi-Bezem-Coquand functional in [4].

Alternatively, one can justify the existence of backward recursion in continuous models by showing that backward recursion is definable from the slightly more general schema of open recursion on the lexicographic ordering considered in [5]. Open recursion is defined to be the fixpoint of the following recursive equation:

$$\text{Open}_{\rho,\tau}^\psi(u) =_{\tau} \psi_u(\lambda n, v . \text{Open}^\psi([u](n) \oplus v) \text{ if } n \in \text{dom}(v) \setminus \text{dom}(u))$$

where once again τ is discrete, and lexicographically open recursive functionals of the above form are shown to be total in [5, Proposition 5.1].

Proposition 3.8. *BkR is instance-wise primitive recursively definable from Open.*

Proof. Primitive recursively define

$$m_{n,u,v} := \text{least } i \leq n \text{ s.t. } i \in \text{dom}(v) \setminus \text{dom}(u), \text{ else } n,$$

and set $\text{BkR}_{\rho,\tau}^\psi(u) = \text{Open}_{\rho,\tau}^{\tilde{\psi}}(u)$ where

$$\tilde{\psi}_u(f^{\mathbb{N} \times \bar{\rho}^{\mathbb{N}} \rightarrow \tau}) := \psi_u(\lambda n, v . f(m_{n,u,v}, u \oplus v) \text{ if } n \in \text{dom}(v) \setminus \text{dom}(u)).$$

Then expanding definitions we have

$$\begin{aligned} \text{BkR}^\psi(u) &= \tilde{\psi}_u(\lambda n, v . \text{BkR}^{\tilde{\psi}}([u](n) \oplus v) \text{ if } n \in \text{dom}(v) \setminus \text{dom}(u)) \\ &\stackrel{(a)}{=} \psi_u(\lambda n, v . \text{BkR}([u](m_{n,u,v}) \oplus (u \oplus v)) \text{ if } n \in \text{dom}(v) \setminus \text{dom}(u)) \\ &\stackrel{(b)}{=} \psi_u(\lambda n, v . \text{BkR}(u \oplus v) \text{ if } n \in \text{dom}(v) \setminus \text{dom}(u)) \end{aligned}$$

where for (a) we use $n \in \text{dom}(v) \setminus \text{dom}(u) \rightarrow m_{n,u,v} \in \text{dom}(u \oplus v) \setminus \text{dom}(u)$, and (b) follows by minimality of $m_{n,u,v}$. \square

Remark 3.9 (Update recursion). It is a fairly easy observation that update recursion as defined in [5] -

$$\text{UpR}^H(u) = H_u(\lambda n, x^\rho . \text{UpR}^H(u_n^x) \text{ if } n \notin \text{dom}(u)),$$

is a simple instance of backward recursion obtained by setting

$$\psi_u(f^{\mathbb{N} \times \bar{\rho}^{\mathbb{N}} \rightarrow \tau}) := H_u(\lambda n, x . f(n, u_n^x) \text{ if } n \notin \text{dom}(u)).$$

Remark 3.10 (Bar recursion). For those readers interested in the computability theory of bar recursion, it might be instructive to pause for a moment to consider a natural instance of bar recursion that arises from backward recursion. Let us define $\text{BarR}(H, s^{\rho^*}) := \text{BkR}^\psi(\hat{s})$ where

$$\psi_u(f^{\mathbb{N} \times \bar{\rho}^{\mathbb{N}} \rightarrow \tau}) :=_{\tau} H(u, \lambda t . f(|t| - 1, \hat{t})).$$

Then it is not too hard to show that BarR satisfies

$$\text{BarR}(H, s) = H(\hat{s}, \lambda t . \text{BarR}(H, s \oplus t) \text{ if } |t| > |s|),$$

and this can be viewed as a ‘implicitly well-founded’ variant of Spector’s bar recursion (1). The reason we highlight this is that while several such implicit variants of Spector’s so-called ‘special’ instance of bar recursion² have been studied, including both modified bar recursion and the implicit product of selection functions, constructing a direct analogue to the general form is more complicated (for example, an implicit form of the so-called product of quantifiers is known not to exist [14]).

The subtle reason for this is that such variants of bar recursion must not be allowed to access the length of the input sequence s . For example, no object Φ can satisfy the slightly altered equation

$$\Phi(H, s) = H(s, \lambda t . \Phi(H, s \oplus t) \text{ if } |t| > |s|)$$

²See [21] for the distinction between the special and general forms of Spector’s bar recursor.

even for discrete output type, since we could just take $\tau = \mathbb{N}$ and define $H(s, g) := 1 + f(s * 0)$, and then $\Phi(\langle \rangle) = n + 1 + \Phi(H, [0](n + 1)) > n$ for all n , which cannot hold in any model of arithmetic. Indeed, trying to define this from BkR with continuous ψ_u is impossible, since we'd require a non-continuous unbounded search (and thus totality of the underlying instance of BkR would no longer be an open predicate). Thus we overcome the difficulty with implicit variants of bar recursion by removing access to the length of the input. Note that this problem is not a feature of modified bar recursion and implicit products of selection functions (or indeed any of the realizers we define in the following sections), since these are defined 'pointwise', and make recursive calls only when we are accessing points already greater than the length of the input sequence.

4 A computational interpretation of the double negation shift

Now that we have completed the mathematical groundwork we come to the core of the paper. In this section we give a new, general realizability interpretation to the double negation shift. Ultimately, this will form a special case of the interpretation of full dependent choice given in the next section. However, by focusing first on the double negation shift we have an opportunity to present our main ideas in a slightly simplified setting, then the extension to full dependent choice will mostly be a matter of taking care of some additional technical details.

4.1 Modified realizability interpretation of extensions of PA^ω

We begin by very briefly recalling how Kreisel's modified realizability can be used in conjunction with the so-called Friedman trick to extract programs from classical proofs of Π_2^0 -formulas. This is all completely standard, so we omit most of the details. For every formula in the language of HA^ω the realizability relation $x \text{ mr } A$ is inductively defined by

$$\begin{aligned} () \text{ mr } A &\equiv A \text{ if } A \text{ is an atomic formula,} \\ x, y \text{ mr } (A \wedge B) &\equiv x \text{ mr } A \wedge y \text{ mr } B, \\ n^{\mathbb{N}}, x, y \text{ mr } (A \vee B) &\equiv (n = 0 \rightarrow x \text{ mr } A) \wedge (n \neq 0 \rightarrow y \text{ mr } B), \\ f \text{ mr } (A \rightarrow B) &\equiv \forall x (x \text{ mr } A \rightarrow f x \text{ mr } B), \\ x \text{ mr } \forall z A(z) &\equiv \forall z (x z \text{ mr } A(z)), \\ x, y \text{ mr } \exists z A(x) &\equiv y \text{ mr } A(x). \end{aligned}$$

It is well-known that whenever $\text{HA}^\omega \vdash A$ then $\text{HA}^\omega \vdash t \text{ mr } A$ where t is some primitive recursive term extracted from the proof of A . The interpretation of classical logic, on the other hand, is more subtle. A simple combination of the negative translation with modified realizability fails to work since the atomic formula \perp is realized by $()$ and therefore all negated formulas are trivially interpreted. In particular, this method gives us no way of extracting realizers from Π_2^0 -formulas $\forall x^{\mathbb{N}} \exists y^{\mathbb{N}} A(x, y)$.

One well established way of overcoming this problem is to slightly alter the definition of modified realizability by regarding $x^{\mathbb{N}} \text{ mr } \perp$ as an uninterpreted formula. Then, as discussed in e.g. [6, 8], from a classical derivation $\text{PA}^\omega \vdash \forall y^{\mathbb{N}} \exists x^{\mathbb{N}} A(y, x)$ one can extract a term t such that $\text{HA}^\omega \vdash \forall y A(y, ty)$, utilising the aforementioned Friedman trick of replacing the formula $x \text{ mr } \perp$ by the quantifier-free formula $A(y, x)$. This idea can be smoothly expanded to extensions of PA^ω with some additional axiom(s) Γ . Provided that $\text{HA}^\omega + \Delta \vdash \Phi \text{ mr } \Gamma^N$, where Γ^N denotes the negative translation of Γ , Δ is some set of axioms satisfying some natural closure properties with respect to \perp , and Φ is some closed term in the language of $\text{HA}^\omega + \Delta$, then from a classical proof $\text{PA}^\omega + \Gamma \vdash \forall y \exists x^{\mathbb{N}} A(y, x)$ one can extract a term t in Φ such that $\text{HA}^\omega + \Delta \vdash \forall y A(y, ty)$.

Thus we have a method that allows us to extract realizers for Π_2^0 formulas from any extension Γ of Peano arithmetic whenever we can realize the negative interpretation Γ^N of Γ . In the remainder of this paper we develop this idea and focus on constructing terms Φ such that $\Phi \text{ mr } \Gamma^N$ where Γ is either countable or countable dependent choice, and Γ^N is the adapted realizability interpretation which treats $x \text{ mr } \perp$ as a new predicate in x . In fact, following [15] we generalise slightly and replace \perp by some arbitrary formula R whose type of realizers is a discrete type, emphasising the fact that \perp acts as some undefined object to be realized. However, if the reader prefers they can just treat this as a relabelling and imagine $R = \perp$ throughout.

4.2 The J -shift and its variants

The axiom of countable choice is given by

$$\text{AC} : \forall n \exists x^\rho B_n(x) \rightarrow \exists \alpha^{\mathbb{N} \rightarrow \rho} \forall n B_n(\alpha(n)).$$

It is well-known that the negative translation of AC,

$$\forall n ((\exists x B_n^N(x) \rightarrow R) \rightarrow R) \rightarrow (\exists \alpha \forall n B_n^N(\alpha(n)) \rightarrow R) \rightarrow R,$$

(here with an arbitrary discretely-realized R in place of \perp) is provable using the (trivially realized) intuitionistic axiom of choice from the simpler double-negation shift,

$$\text{DNS} : \forall n ((A(n) \rightarrow R) \rightarrow R) \rightarrow (\forall n A(n) \rightarrow R) \rightarrow R,$$

by setting $A(n) := \exists x B_n^N(x)$. Thus a realizability interpretation of countable choice follows directly from that of DNS. Note that this version of DNS for arbitrary R is also called the K -shift in [15].

In order to successfully realize DNS one typically relies on a term h realizing ex-falso-quodlibet in the form $\forall n (R \rightarrow A(n))$, and so in practice one must work with a restricted class of formulas $A(n)$ that admit such a realizer, such as any formula in the image of the negative translation (for which one can trivially construct such a h even uniformly in n). This need for additional realizers and a corresponding restriction on formulas can seem slightly inelegant, and so a reformulation of DNS is given in [15] which, rather than separately assuming $R \rightarrow A(n)$, adds this positive information directly to the premise of DNS, yielding

$$\forall n ((A(n) \rightarrow R) \rightarrow A(n)) \rightarrow (\forall n A(n) \rightarrow R) \rightarrow R.$$

In [15] this is actually written in a equivalent form called the J -shift:

$$J\text{-shift} : \forall n ((A(n) \rightarrow R) \rightarrow A(n)) \rightarrow (\forall n A(n) \rightarrow R) \rightarrow \forall n A(n),$$

and this is given a realizability interpretation using the product of selection functions, an interpretation which is valid for *arbitrary* formulas $A(n)$. Then, in the case that $R \rightarrow A(n)$ is realizable, one easily reobtains an interpretation of the double negation shift:

Proposition 4.1 ([15]). *J -shift implies DNS over minimal logic, whenever $R \rightarrow A(n)$ holds.*

Here, we adopt the convention of [15] in adding the positive information we need directly to the premise of the double negation shift, so that our interpretation is valid for all $A(n)$, and as in [15], and as we show in our examples in Section 5.2, we can always convert our realizer to one of DNS for negated formulas $A(n)$, which in turn is sufficient to realize the axiom of countable choice.

For notational reasons we interpret a pair of syntactically more flexible variants of the J -shift, designed to match the family of realizers we construct.

Definition 4.2. We define the J_i^* -shifts for $i = 1, 2$ by

$$\begin{aligned} J_1^*\text{-shift} & : \forall m, n ((A(m) \rightarrow R) \rightarrow A(n)) \rightarrow (\forall n A(n) \rightarrow R) \rightarrow R \\ J_2^*\text{-shift} & : \forall m, n ((A(m) \rightarrow R) \rightarrow A(n)) \rightarrow (\forall n A(n) \rightarrow R) \rightarrow \forall n A(n) \end{aligned}$$

where R has discrete realizing type.

The following result confirms that our J_i^* -shift principles are nothing more than simple rephrasings of the original J -shift.

Proposition 4.3. *J_1^* -shift $\Leftrightarrow J_2^*$ -shift $\Leftrightarrow J$ -shift over minimal logic.*

Proof. J_1^* -shift $\Rightarrow J_2^*$ -shift follows from the observation that

$$\forall m, n ((A(m) \rightarrow R) \rightarrow A(n)) \rightarrow (R \rightarrow \forall n A(n)),$$

which is true because for arbitrary n we have

$$R \rightarrow (A(n) \rightarrow R)$$

and thus

$$((A(n) \rightarrow R) \rightarrow A(n)) \rightarrow (R \rightarrow A(n)).$$

The implication J_2^* -shift \Rightarrow J -shift follows from

$$\forall n((A(n) \rightarrow R) \rightarrow A(n)) \rightarrow \forall m, n((A(m) \rightarrow R) \rightarrow A(n)),$$

which is true because firstly

$$((A(m) \rightarrow R) \rightarrow A(m)) \rightarrow ((A(m) \rightarrow R) \rightarrow R),$$

and since $R \rightarrow (A(n) \rightarrow R)$ this yields

$$((A(m) \rightarrow R) \rightarrow A(m)) \rightarrow ((A(m) \rightarrow R) \rightarrow (A(n) \rightarrow R))$$

and so finally

$$((A(n) \rightarrow R) \rightarrow A(n)) \wedge ((A(m) \rightarrow R) \rightarrow A(m)) \rightarrow ((A(m) \rightarrow R) \rightarrow A(n))$$

The remaining direction J -shift \Rightarrow J_1^* -shift is straightforward. \square

The reasons that we highlight these variants of the J -shift is that we want versions of the J -shift that are directly realized by our variants of bar recursion. As we will see, in most cases each variant will only use the premise of J_i^* -shift for $(m, n) \in I \subseteq \mathbb{N} \times \mathbb{N}$ for some I .

4.3 Realizing the J_i^* -shift

We focus on constructing a realizer for the J_1^* -shift, then a realizer of the J_2^* -shift comes out immediately. Suppose that the realizing types of $A(n)$ and R are ρ and τ respectively, where we assume that τ is discrete. The J_1^* -shift is realized by a term Φ of type $(\mathbb{N} \rightarrow \mathbb{N} \rightarrow (\rho \rightarrow \tau) \rightarrow \rho) \rightarrow (\rho^{\mathbb{N}} \rightarrow \tau) \rightarrow \tau$, which, given terms $\varepsilon: \mathbb{N} \rightarrow \mathbb{N} \rightarrow (\rho \rightarrow \tau) \rightarrow \rho$ and $q: \rho^{\mathbb{N}} \rightarrow \tau$ that satisfy

$$\begin{aligned} \forall m, n, p^{\rho \rightarrow \tau} (\forall x^{\rho} (x \text{ mr } A(m) \rightarrow p(x) \text{ mr } R) \rightarrow \varepsilon_{m,n}(p) \text{ mr } A(n)) \\ \forall \alpha^{\rho^{\mathbb{N}}} (\forall n (\alpha(n) \text{ mr } A(n)) \rightarrow q(\alpha) \text{ mr } R) \end{aligned}$$

returns a term $\Phi \varepsilon q: \tau$ satisfying $\Phi \varepsilon q \text{ mr } R$.

The basic idea that unites all such existing realizers of double-negation shift principles is to form an auxiliary functional Ψ which performs a backward recursive loop, which builds increasingly large partial realizers of $\forall n A(n)$.

More precisely, suppose that ε and q satisfy the premise of the J_1^* shift as above, and imagine we are given a partial realizer $u: \rho^{\mathbb{N}}$ which satisfies $\forall n \in \text{dom}(u) (u(n) \text{ mr } A(n))$. Then let us somewhat informally define $\Psi^{\varepsilon, q} u := q(u \oplus c(u))$, where c is some as yet unspecified function on partial sequences, but the aim is that it forms a completion of the partial realizer of u and that therefore $\Psi^{\varepsilon, q} u \text{ mr } R$ for all u . This completion $c(u)$ can be constructed by backward recursion using ε . For $n \notin \text{dom}(u)$ we define

$$c(u)(n) := \varepsilon_{mnu, n}(\lambda x. \Psi^{\varepsilon, q}(vnu_m^x))$$

for some index mnu and partial realizer vnu of $\forall n A(n)$. Again, both mnu and vnu are left unspecified for now, but note that if at the very least

- (i) vnu is a partial realizer of $\forall n A(n)$ satisfying $u \sqsubseteq vnu$ and
- (ii) $mnu \notin \text{dom}(u)$ whenever $n \notin \text{dom}(u)$,

then $u \sqsubseteq vnu_m^x$ and vnu_m^x is a partial realizer whenever $x \text{ mr } A(mnu)$. Therefore $\lambda x. \Psi^{\varepsilon, q}(vnu_m^x)$ is a realizer of $A(mnu) \rightarrow R$ under the assumption that Ψ is correct for all *extensions* of u . The idea now is that we can admit this assumption as a backward induction hypothesis, and so by backward induction we can prove that $\Psi^{\varepsilon, q} u \text{ mr } R$ for all u . Then setting $\Phi \varepsilon q := \Psi^{\varepsilon, q} \emptyset$ gives us a realizer for the J_1^* -shift, since \emptyset is trivially a partial realizer of $\forall n A(n)$.

What remains is to formalise this idea and make some sensible choice of mnu and vnv satisfying (i) and (ii) above. The most natural might be to set $mnu = n$ and $vnv = u$ - in this case, as we will see below, is precisely the idea behind the Berardi-Bezem-Coquand functional of [3]. However, more intricate choices lead to other realizers, including modified bar recursion and the product of selection functions. Our aim now is to make this intuition precise, and provide a sufficiently rich general construction of m and v which captures all of these realizers and much more.

Proposition 4.4. *Suppose that ρ and τ are the realizing types of $A(n)$ and R , with τ discrete, and that we are given a computable relation $\prec: \bar{\rho}^{\mathbb{N}} \rightarrow (\mathbb{N} \times \mathbb{N} \rightarrow \mathbb{B})$ and an index $m: \mathbb{N} \times \bar{\rho}^{\mathbb{N}} \rightarrow \mathbb{N}$ such that*

(i) \prec_u is well-founded,

(ii) $n \notin \text{dom}(u) \rightarrow mnu \notin \text{dom}(u) \cup \{k \mid k \prec_u n\}$

for all u and n . Then there is a term $\Psi_{(\prec, m)}^{\varepsilon, q}: \bar{\rho}^{\mathbb{N}} \rightarrow \tau$ with parameters of type $\varepsilon: \mathbb{N} \rightarrow \mathbb{N} \rightarrow (\rho \rightarrow \tau) \rightarrow \rho$ and $q: \rho^{\mathbb{N}} \rightarrow \tau$ which is primitive recursively definable in $\text{BkR} + \lambda u. \text{wR}_{\prec_u}$, and satisfies the recursive equation

$$\Psi^{\varepsilon, q}(u) = q\left(u \oplus \underbrace{\lambda n. \varepsilon_{mnu, n}(\lambda x. \Psi^{\varepsilon, q}(u \oplus \langle \alpha_u \rangle(n)_{mnu}^x))}_{\alpha_u}\right),$$

where $\alpha_u: \rho^{\mathbb{N}}$ denotes the argument of q as indicated above, and $\langle \alpha_u \rangle(n): \bar{\rho}^{\mathbb{N}}$ is defined to be

$$\lambda k. (\alpha_u(k) \text{ if } k \prec_u n \text{ else } \perp).$$

Remark 4.5. Technically speaking the term $\lambda u. \text{wR}_{\prec_u}$ is not properly defined in E-HA^ω - we are simply assuming here that there exists a function F definable in E-HA^ω such that for all u , $F(u)$ satisfies the defining equation of wR_{\prec_u} i.e. well-founded recursion over \prec_u as considered in Section 3.4. However, in all of the concrete examples we consider, \prec will not depend on u and wR_{\prec} will always be trivially definable in E-HA^ω , so this rather casual definition will not be problematic.

Proposition 4.4 above is a special case of Proposition 5.2 in the next section, whose proof can be found in the appendix, and so we omit a proof of Proposition 4.4 here. However, on an informal level, it is not too difficult to see that Ψ is well-defined. First we note that the definition of $\langle \alpha_u \rangle(n)$ is not circular, since it is only used to define $\alpha_u(n)$, and thus α_u as a whole is constructed using the well-founded recursor wR_{\prec_u} . Then one observes that the whole expression is a well-defined backward recursive functional since to compute $\Psi(u)$ we only call Ψ on arguments of the form $u \oplus \langle \alpha_u \rangle(n)_m^x$ for $m \notin \text{dom}(u)$ by condition (ii), which are always strict extensions of u .

Theorem 4.6. *Suppose that $\Psi_{(\prec, m)}$ is defined as in Proposition 4.4 for \prec and m satisfying conditions (i) and (ii). Then the term $\Phi_{(\prec, m)} := \lambda \varepsilon, q. \Psi_{(\prec, m)}^{\varepsilon, q}(\emptyset)$ realizes the J_1^* -shift, provably in $\text{E-HA}^\omega + \text{Cont} + \text{Bkl} + \text{BkR} + (\text{wl}_{\prec_u}) + \lambda u. \text{wR}_{\prec_u}$.*

Remark 4.7. Here (wl_{\prec_u}) denotes the collection of well-founded induction schemata over the well-founded relations \prec_u . As with the corresponding modes of recursion discussed in Remark 4.5, in practise \prec_u will typically not depend on u and wl_{\prec} will be easily provable in E-HA^ω .

Proof. Assume that ε and q realize the premise of the J_1^* -shift. We prove that $\Phi_{(\prec, m)} \varepsilon q \text{ mr } R$ using a main backward induction and an auxiliary well-founded induction. Let us define

$$B(u) := \forall n \in \text{dom}(u) (u(n) \text{ mr } A(n)) \rightarrow \Psi^{\varepsilon, q}(u) \text{ mr } R.$$

This is an open formula in the sense of Proposition 3.2 since by Cont we have

$$\Psi(u) \text{ mr } R \leftrightarrow \exists n \forall w \Psi([u](n) \oplus w) \text{ mr } R.$$

Now, to prove the backward induction step for B , assume that u is a partial realizer of $\forall n A(n)$ (i.e. the premise of $B(u)$ holds) and suppose that $B(v)$ holds for all $v \sqsupset u$. We want to show that $\Psi^{\varepsilon, q}(u) \text{ mr } R$.

We do this by first proving using wl_{\prec_u} that α_u (as defined in Proposition 4.4) realizes $\forall n A(n)$. Fix n and assume as an auxiliary induction hypothesis that $\alpha_u(k) \text{ mr } A(k)$ for all $k \prec_u n$. If $n \in \text{dom}(u)$ we trivially have $\alpha_u(n) = u(n) \text{ mr } A(n)$, so assume that $n \notin \text{dom}(u)$. In this case, first observe that

$$x \text{ mr } A(mnu) \rightarrow \forall i \in \text{dom}(v) (v(i) \text{ mr } A(i)) \quad \text{and} \quad \forall i \in \text{dom}(v) (v(i) \text{ mr } A(i)) \rightarrow \Psi^{\varepsilon, q}(v) \text{ mr } R$$

for $v := u \oplus \langle \alpha_u \rangle(n)_{mnu}^x$. The first step is clear by the auxiliary induction hypothesis, which implies that $u \oplus \langle \alpha_u \rangle(n)$ is a partial realizer of $\forall n A(n)$, while the second step follows from the main hypothesis $B(v)$, since by condition (ii) we know that $n \notin \text{dom}(u)$ implies that $mnu \notin \text{dom}(u)$ and thus $v \sqsupset u$. Putting this together we see that $\lambda x. \Psi(u \oplus \langle \alpha_u \rangle(n)_{mnu}^x) \text{ mr } (A(mnu) \rightarrow R)$ and thus

$$\alpha_u(n) = \varepsilon_{mnu,n}(\lambda x. \Psi(u \oplus \langle \alpha_u \rangle(n)_{mnu}^x) \text{ mr } A(n)).$$

by correctness of ε . This completes the auxiliary well-founded induction, giving us $\forall n(\alpha_u(n) \text{ mr } A(n))$, and therefore $\Psi(u) = q(\alpha_u) \text{ mr } R$, which completes the main backward induction step. Finally, then, we obtain $\forall u B(u)$ by **Bkl**, and so in particular by $B(\emptyset)$ we have $\Psi^{\varepsilon,q}(\emptyset) \text{ mr } R$, which completes the proof. \square

Corollary 4.8. *There is a term $\tilde{\Phi}_{\langle \cdot, m \rangle}$ primitive recursive in $\text{BkR} + \lambda u. \text{wR}_{\prec_u}$ which realizes the J_2^* -shift, provably in $\text{E-HA}^\omega + \text{Cont} + \text{Bkl} + \text{BkR} + (\text{wl}_{\prec_u}) + \lambda u. \text{wR}_{\prec_u}$.*

Proof. Keeping all the notation of Theorem 4.6, define $\tilde{\Psi}^{\varepsilon,q}(u) = \alpha_u$ so that it satisfies the recursive equation

$$\tilde{\Psi}^{\varepsilon,q}(u) = u \oplus \lambda n . \varepsilon_{mnu,n}(\lambda x . q(\tilde{\Psi}^{\varepsilon,q}(u \oplus \langle \alpha_u \rangle(mnu)_{mnu}^x))).$$

Define $\tilde{\Phi}_{\varepsilon q} := \tilde{\Psi}^{\varepsilon,q}(\emptyset)$. Then it follows immediately from $\forall u B(u)$ and \prec_\emptyset -induction, as in the proof of Theorem 4.6, that $\tilde{\Phi}_{\varepsilon q} \text{ mr } \forall n A(n)$. \square

Let us now briefly consider some specific instantiations of the parameters of Theorem 4.6 (a more detailed discussion will be given in Section 5.2). Firstly, setting $\Psi_0 = \Psi_{\langle \cdot, m \rangle}$ in the simple case that $\prec_u = \emptyset$ and $mnu = n$ for all u , we have $\langle \alpha_u \rangle(n) = \emptyset$ and, using the abbreviation ε_n for $\varepsilon_{n,n}$ our realizer becomes

$$\Psi_0^{\varepsilon,q}(u) = q(u \oplus \lambda n . \varepsilon_n(\lambda x . \Psi_0(u_n^x)))$$

which is nothing more than a simple variant of the Berardi-Bezem-Coquand realizer of countable choice given in [3] and discussed in [5]. On the other hand, suppose that we still keep $\prec_u = \emptyset$ for all u , but define $mnu := \mu i \leq n (i \notin \text{dom}(u))$, where recall that μ is the bounded search operator which in this case returns the least $i \leq n$ satisfying $i \notin \text{dom}(u)$, and n if no such i exists. Then finite input for this variant of Ψ will be of the form \hat{s} for some sequences $s : \rho^*$. Defining $\Psi_1(s) := \Psi_{(\emptyset, m)}(\hat{s})$ and observing that for $n \geq |s|$ we have $m\hat{s} = |s|$, we obtain

$$\Psi_1^{\varepsilon,q}(s) := q(\hat{s} \oplus \lambda n . \varepsilon_{|s|,n}(\lambda x . \Psi_1(s * x)))$$

which is just a (non-dependent) form of modified bar recursion. Finally, let us define $\prec_u = <$ and $mnu = n$ for all u . Then setting $\Psi_2(s) := \Psi_{\langle \cdot, m \rangle}(\hat{s})$, abbreviating $\varepsilon_{n,n}$ by ε_n and observing that for $n \geq |s|$ we have $\hat{s} \oplus \langle \alpha_u \rangle(n) = [\alpha_u](n)$ we obtain a realizing term satisfying

$$\Psi_2^{\varepsilon,q}(s) = q(\underbrace{\hat{s} \oplus \lambda n . \varepsilon_n(\lambda x . \Psi_2([\alpha_u](n) * x))}_{\alpha_u}).$$

The corresponding variant $\tilde{\Psi}_2$ which realizes J_2^* -shift is exactly the simple implicit product of selection functions of [12].

Thus three completely different modified realizability interpretations of countable choice appear as simple instances of Theorem 4.6. Moreover, in each instance we only require a restricted form of backward induction and well-founded recursion which corresponds exactly to the soundness proofs used in the original papers: for Ψ_0 Theorem 4.6 is reduced to the proof of the double negation shift using update recursion given in [5], while Ψ_1 and Ψ_2 require backward induction relativised to downward closed partial functions, which is entirely equivalent to the variants of bar induction used to prove their correctness in [6] and [12] respectively. Thus Theorem 4.6 doesn't simply provide a parametrised framework with which different realizers can be compared, but also a framework in which their correctness proofs can be viewed in a uniform way as relativisations of backward induction.

Of course the construction of such a framework is only partially motivated by the desire to compare existing interpretations. Theorem 4.6 generalises existing work in that one can use an arbitrary parameters to define new realizers of the J_i^* -shift that are automatically correct, giving an additional level of flexibility and power when it comes to extracting computational content from proofs in practise.

However, we do not discuss this in any more detail here, instead proceeding straight to the generalisation of Theorem 4.6 to full dependent choice.

5 A computational interpretation of dependent choice

We now give a parametrised realizability interpretation to the principle of countable dependent choice. Our formulation of dependent choice will be slightly more general than the usual sequential variants treated in e.g. [6, 15, 26], in the sense that we parametrise the principle itself by a decidable well-founded relation \triangleleft on \mathbb{N} which dictates the underlying dependency of the choice sequence.

To be more precise, given a decidable strict well-founded partial order \triangleleft let us extend our type system with types ρ^{\triangleleft} which represent the set $\bigcup_{n \in \mathbb{N}} \rho^{\triangleleft n}$ where

$$\rho^{\triangleleft n} := \{m \mid m \triangleleft n\} \rightarrow \rho.$$

We tacitly assume that the types ρ^{\triangleleft} can be smoothly incorporated into our system, and come equipped with a length function $|\cdot|: \rho^{\triangleleft} \rightarrow \mathbb{N}$ returning for each $t: \rho^{\triangleleft}$ a unique index $|t|$ such that $t \in \rho^{\triangleleft |t|}$. For $\triangleleft = <$, the type ρ^{\triangleleft} is isomorphic to the type ρ^* of finite sequences over ρ , objects of type

$$\{m \mid m < n\} \rightarrow \rho$$

representing finite sequences of length n . In fact ρ^{\triangleleft} is essentially a generalisation of the finite sequence type to arbitrary \triangleleft -closed partial functions (which need not have finite domain, though). Note that when $\triangleleft = \emptyset$ is the empty relation, the type ρ^{\emptyset} is isomorphic to \mathbb{N} , and as we will see, in this case our parametrised dependent choice principle collapses to normal countable choice.

Now, the principle of \triangleleft -DC_{seq} is given by the schema

$$\forall s^{\rho^{\triangleleft}} (\forall i \triangleleft |s| A_i(\{s\}(i), s(i)) \rightarrow \exists x^{\rho} A_{|s|}(s, x)) \rightarrow \exists \alpha^{\mathbb{N} \rightarrow \rho} \forall n A_n(\{\alpha\}(n), \alpha(n))$$

where $\{\alpha\}(n): \rho^{\triangleleft n}$ is just the \triangleleft -initial segment of α i.e. $\lambda m \triangleleft n. \alpha(m)$, and analogously for s (note that $\{s\}(i)$ is well-defined for $i \triangleleft |s|$ by transitivity of \triangleleft). Note that \triangleleft -DC_{seq} follows from the full axiom of choice together with well-founded recursion over \triangleleft as follows: by classical logic and full choice we have

$$\forall s (\forall i \triangleleft |s| A_i(\{s\}(i), s(i)) \rightarrow \exists x A_{|s|}(s, x)) \rightarrow \exists \Theta \forall s (\forall i \triangleleft |s| A_i(\{s\}(i), s(i)) \rightarrow A_{|s|}(s, \Theta s)).$$

Now, recursively defining $\alpha(n) := \Theta(\{\alpha\}(n))$, we prove $\forall n A_n(\{\alpha\}(n), \alpha(n))$ by $\text{wl}_{\triangleleft}$, since from the assumption that $\forall i \triangleleft n A_i(\{\alpha\}(i), \alpha(i))$ we obtain $A_n(\{\alpha\}(n), \alpha(n))$ using that fact that $\{\{\alpha\}(n)\}(i) = \{\alpha\}(i)$ for $i \triangleleft n$.

In Section 5.2 below we discuss more well-known instances of dependent choice, including its canonical formulation as

$$\forall n, x^X \exists y A_n(x, y) \rightarrow \exists f \forall n A_n(f(n), f(n+1)),$$

which is easily provable from $<$ -DC_{seq}. However, here we take advantage of the fact that our setting allows us to interpret dependent choice in the non-standard, but very general form \triangleleft -DC_{seq}.

5.1 Realizing \triangleleft -DC_{seq}^{J_i*}

As in the previous section, we realize a positive form of the negative translation of \triangleleft -DC_{seq} which is somewhat analogous to the dependent J -shift of [15]. As with the previous section, we can easily derive a realizer for the standard negative translation of dependent choice principles from a realizer of our shift principles - and we illustrate this in Section 5.2.

Definition 5.1. We define the translated principle \triangleleft -DC_{seq}^{J_i*} by

$$\begin{aligned} \triangleleft\text{-DC}_{\text{seq}}^{J_1^*} &: \left\{ \begin{array}{l} \forall s, r (\forall i \triangleleft |s| A_i(\{s\}(i), s(i)) \rightarrow (\exists x A_{|s|}(s, x) \rightarrow R) \rightarrow \exists x A_{|r|}(r, x)) \\ \rightarrow (\exists \alpha \forall n A_n(\{\alpha\}(n), \alpha(n)) \rightarrow R) \rightarrow R \end{array} \right. \\ \triangleleft\text{-DC}_{\text{seq}}^{J_2^*} &: \left\{ \begin{array}{l} \forall s, r (\forall i \triangleleft |s| A_i(\{s\}(i), s(i)) \rightarrow (\exists x A_{|s|}(s, x) \rightarrow R) \rightarrow \exists x A_{|r|}(r, x)) \\ \rightarrow (\exists \alpha \forall n A_n(\{\alpha\}(n), \alpha(n)) \rightarrow R) \rightarrow \exists \alpha \forall n A_n(\{\alpha\}(n), \alpha(n)) \end{array} \right. \end{aligned}$$

where A is arbitrary formula over $\mathbb{N} \times \rho^{\triangleleft} \times \rho$ and the realizing type of R is restricted to being discrete.

Again, these variants are nothing more than convenient syntactical rephrasings of the dependent shift principle given in [15], this time extended to arbitrary partial orderings \triangleleft .

Now, if σ and τ are the realizing types of $A_n(s, x)$ and R respectively, then $\triangleleft\text{-DC}_{\text{seq}}^{J_1^*}$ is realized by a term Φ of type $((\rho \times \sigma)^{\triangleleft} \rightarrow \rho^{\triangleleft} \rightarrow (\rho \times \sigma \rightarrow \tau) \rightarrow \rho \times \sigma) \rightarrow ((\rho \times \sigma)^{\mathbb{N}} \rightarrow \tau) \rightarrow \tau$ which given input $\varepsilon: (\rho \times \sigma)^{\triangleleft} \rightarrow \rho^{\triangleleft} \rightarrow (\rho \times \sigma \rightarrow \tau) \rightarrow \rho \times \sigma$ and $q: (\rho \times \sigma)^{\mathbb{N}} \rightarrow \tau$ that satisfies

$$\forall s^{(\rho \times \sigma)^{\triangleleft}}, r^{\rho^{\triangleleft}}, p^{\rho \times \sigma \rightarrow \tau} \left\{ \begin{array}{l} \forall i \triangleleft |s|(s(i)_1 \text{ mr } A_i(\{s_0\}(i), s(i)_0)) \rightarrow \\ (\forall x^{\rho \times \sigma} (x_1 \text{ mr } A_{|s|}(s_0, x_0) \rightarrow p(x) \text{ mr } R) \rightarrow \varepsilon_{s,r}(p)_1 \text{ mr } A_{|r|}(r, \varepsilon_{s,r}(p)_0)) \end{array} \right.$$

$$\forall \alpha^{(\rho \times \sigma)^{\mathbb{N}}} (\forall n \alpha(n)_1 \text{ mr } A_n(\{\alpha_0\}(n), \alpha(n)_0) \rightarrow q(\alpha) \text{ mr } R)$$

returns a term $\Phi \varepsilon q: \tau$ satisfying $\Phi \varepsilon q \text{ mr } R$. In the formulae above and results that follows, $s_0: \rho^{\triangleleft}$ denotes the first projection of the term $s: (\rho \times \sigma)^{\triangleleft}$ i.e. $s_0(m) := s(m)_0$, and similarly for infinite sequences α . As in the previous section, we construct a family of realizing terms which follow the same basic principle of backward recursion on partial functions, and which give a computational interpretation to $\triangleleft\text{-DC}_{\text{seq}}^{J_1^*}$. However, this time our choices of \prec_u and mnu require some additional conditions to ensure they are now compatible with \triangleleft .

Proposition 5.2. *Suppose that σ and τ are the realizing types of $A(n)$ and R , with τ discrete, and that $\prec: \overline{(\rho \times \sigma)^{\mathbb{N}}} \rightarrow (\mathbb{N} \times \mathbb{N} \rightarrow \mathbb{B})$ and $m: \mathbb{N} \times \overline{(\rho \times \sigma)^{\mathbb{N}}} \rightarrow \mathbb{N}$ are such that*

- (i) $\forall k(k \prec_u n \vee k \triangleleft n \vee k \triangleleft mnu \rightarrow k \prec'_u n)$ for some well-founded relation \prec'_u ,
- (ii) $n \notin \text{dom}(u) \rightarrow mnu \notin \text{dom}(u) \cup \{k \mid k \prec_u n\}$,

for all u and n . Then there is a term $\Psi_{(\triangleleft, \prec, m)}^{\varepsilon, q}: \overline{(\rho \times \sigma)^{\mathbb{N}}} \rightarrow \tau$ with parameters of type $\varepsilon: ((\rho \times \sigma)^{\triangleleft} \rightarrow \rho^{\triangleleft} \rightarrow (\rho \times \sigma \rightarrow \tau) \rightarrow \rho \times \sigma)$ and $q: ((\rho \times \sigma)^{\mathbb{N}} \rightarrow \tau)$ which is primitive recursively definable in $\text{BkR} + \lambda u.\text{wR}_{\prec'_u}$, and satisfies the recursive equation

$$\Psi^{\varepsilon, q}(u) = \underbrace{q(u \oplus \lambda n . \varepsilon_{\{\alpha\}(mnu), \{\alpha_0\}(n)})(\lambda x . \Psi^{\varepsilon, q}(u \oplus \langle \alpha_u \rangle(n)_{mnu}^x))}_{\alpha_u}$$

where $\alpha_u: (\rho \times \sigma)^{\mathbb{N}}$ denotes the argument of q as indicated above and $\langle \alpha_u \rangle(n): \overline{(\rho \times \sigma)^{\mathbb{N}}}$ is defined to be

$$\lambda k. (\alpha_u(k) \text{ if } k \prec_u n \text{ else } \perp).$$

Regarding the somewhat casual definition of the term $\lambda u.\text{wR}_{\prec'_u}$ see Remark 4.5 - again, in all concrete cases discussed later \prec_u will not even depend on u and $\text{wR}_{\prec'_u}$ will be trivially definable in E-HA^ω . As with Proposition 4.4, on an intuitive level it is not too hard to see that Ψ is well-defined. Firstly α_u is well-defined by \prec'_u -recursion since in order to compute $\alpha_u(n)$ we require $\alpha_u(k)$ for either $k \prec_u n$, $k \triangleleft n$ or $k \triangleleft mnu$, which by condition (i) implies that $k \prec'_u n$. Then by (ii) for $n \notin \text{dom}(u)$ we have $mnu \notin \text{dom}(u)$ and thus $u \sqsubset u \oplus \langle \alpha \rangle(n)_{mnu}^x$ and so recursive calls on Ψ are always made on strict extensions u . A formal proof that Ψ is definable in $\text{BkR} + \lambda u.\text{wR}_{\prec_u}$ is given in Appendix A.

Theorem 5.3. *Suppose that $\Psi_{(\triangleleft, \prec, m)}$ is defined as in Proposition 5.2 for \triangleleft, \prec and m satisfying conditions (i) and (ii), and additionally*

- (iii) if $\text{dom}(u)$ is \triangleleft -closed then so are $\text{dom}(u) \cup \{k \mid k \prec_u n\}$ and $\text{dom}(u) \cup \{k \mid k \prec_u n\} \cup \{mnu\}$,

Then the term $\Phi_{(\triangleleft, \prec, m)} := \lambda \varepsilon, q . \Psi_{(\triangleleft, \prec, m)}^{\varepsilon, q}(\emptyset)$ realizes $\triangleleft\text{-DC}_{\text{seq}}^{J_1^*}$ provably in $\mathcal{T} + \text{Cont} + \text{Bkl} + \text{BkR} + (\text{wl}_{\prec'_u}) + \lambda u.\text{wR}_{\prec'_u}$.

Remark 5.4. While the proof of this Theorem is fairly lengthy, it is little more than a straightforward variant of the much simpler proof of Theorem 4.6. The additional technical details are simply required to deal with the dependency \triangleleft .

Remark 5.5. All terms $s: \rho^{\triangleleft}$, when embedded as partial functions, are \triangleleft -closed by transitivity of \triangleleft , but the converse is not necessarily true.

Proof. Assume that ε and q realize the premise of $\triangleleft\text{-DC}_{\text{seq}}^{J_1^*}$. Analogous to the proof of Theorem 4.6, we use a main backward induction along with an auxiliary well-founded induction, this time on the slightly more complex formula

$$B(u) := \bar{A}(u) \rightarrow \Psi^{\varepsilon, q}(u) \text{ mr } R.$$

for

$$\bar{A}(u) := \forall n \in \text{dom}(u) (\forall k \triangleleft n (k \in \text{dom}(u) \wedge u(n)_1 \text{ mr } A(\{u_0\}(n), u(n)_0)),$$

where now as our premise we require that u is a \triangleleft -closed partial realizer of the dependent choice sequence. As before, $B(u)$ is open in the sense of Proposition 3.2 by **Cont**. To prove the backward induction step, let's assume that $\bar{A}(u)$ holds and $B(v)$ is true for all $v \sqsupset u$ and try to derive $\Psi(u) \text{ mr } R$.

This time we use \prec'_u induction to show that $\forall n C(n)$ where

$$C(n) := \alpha_u(n)_1 \text{ mr } A_n(\{(\alpha_u)_0\}(n), \alpha_u(n)_0).$$

Then we are done since this would imply that $\Psi(u) = q(\alpha_u) \text{ mr } R$. So fix n and assume that $\forall k \prec'_u n C(k)$. There are two cases to consider. If $n \in \text{dom}(u)$ and then by \triangleleft -closedness of u , $C(n)$ becomes $u(n)_1 \text{ mr } A_n(\{u_0\}(n), u(n)_0)$ which is true by assumption. Otherwise, if $n \notin \text{dom}(u)$ then $C(n)$ becomes

$$(*) \quad \varepsilon_{\{\alpha\}(mnu), \{\alpha_0\}(n)}(p)_1 \text{ mr } A_n(\{(\alpha_u)_0\}(n), \varepsilon_{\{\alpha\}(mnu), \{\alpha_0\}(n)}(p)_0)$$

where

$$p := \lambda x . \Psi^{\varepsilon, q}(u \oplus \langle \alpha_u \rangle(n)_{mnu}^x).$$

We now prove that

$$\begin{aligned} x_1 \text{ mr } A_{mnu}(\{(\alpha_u)_0\}(mnu), x_0) &\rightarrow \bar{A}(u \oplus \langle \alpha_u \rangle(n)_{mnu}^x) \text{ and} \\ \bar{A}(u \oplus \langle \alpha_u \rangle(n)_{mnu}^x) &\rightarrow \Psi(u \oplus \langle \alpha_u \rangle(n)_{mnu}^x) \text{ mr } R. \end{aligned}$$

The second implication follows from the main backward induction hypothesis since $u \sqsupset u \oplus \langle \alpha_u \rangle(n)_{mnu}^x$ by (ii). To prove the first implication, let's abbreviate $v := u \oplus \langle \alpha_u \rangle(n)_{mnu}^x$. Then $\text{dom}(v) = \text{dom}(u) \cup \{k \mid k \prec_u n\} \cup \{mnu\}$, and since by $\bar{A}(u)$ we know that $\text{dom}(u)$ is \triangleleft -closed, so is $\text{dom}(v)$ by condition (iii). Thus to obtain $\bar{A}(v)$ it remains to show that

$$(**) \quad \forall k \in \text{dom}(v) (v(k)_1 \text{ mr } A(\{v_0\}(k), v(k)_0)).$$

There are three possibilities. If $k \in \text{dom}(u)$ then $(**)$ follows from $\bar{A}(u)$, while if $k = mnu$ then $(**)$ becomes $x_1 \text{ mr } A(\{(\alpha_u)_0\}(mnu), x_0)$ by (iii) which is our premise. Otherwise, for $k \prec_u n$, $(**)$ becomes $\alpha_u(k)_1 \text{ mr } A(\{(\alpha_u)_0\}(k), \alpha_u(k)_0)$ by condition (iii), and this we know is true by the \prec'_u -induction hypothesis.

Therefore, putting everything together we obtain

$$x_1 \text{ mr } A_{mnu}(\{(\alpha_u)_0\}(mnu), x_0) \rightarrow p(x) \text{ mr } R.$$

Since in addition we have $\forall i \triangleleft mnu (\alpha_u(i)_1 \text{ mr } A_i(\{(\alpha_u)_0\}(i), \alpha_u(i)_0))$, a fact which also follows from the \prec'_u -induction hypothesis along with condition (i), then by correctness of ε we obtain $(*)$.

Thus we have shown that $\forall k \prec'_u n C(k) \rightarrow C(n)$ and therefore $\forall n C(n)$ follows by induction. Since this in turn implies $\Psi(u) \text{ mr } R$, we have shown $\forall v \sqsupset u B(v) \rightarrow B(u)$, and thus $\forall u B(u)$ by backward induction. Finally, then, since $\bar{A}(\emptyset)$ is trivially satisfied, $B(\emptyset)$ implies $\Psi^{\varepsilon, q}(\emptyset) \text{ mr } R$, which completes the proof. \square

Corollary 5.6. *There is a term $\tilde{\Phi}_{(\triangleleft, \prec, m)}$ primitive recursive in $\text{BkR} + \lambda u. \text{wR}_{\prec_u}$ which realizes the $\triangleleft\text{-DC}_{\text{seq}}^{J_2^*}$, provably in $\text{E-HA}^\omega + \text{Cont} + \text{Bkl} + \text{BkR} + (\text{wl}_{\prec_u}) + \lambda u. \text{wR}_{\prec_u}$.*

Proof. Just as in the previous section, and keeping the notation of Theorem 5.3, define $\tilde{\Psi}^{\varepsilon, q}(u) = \alpha_u$ so that it satisfies the recursive equation

$$\tilde{\Psi}^{\varepsilon, q}(u) = u \oplus \lambda n . \varepsilon_{\{\alpha\}(mnu), \{\alpha_0\}(n)}(\lambda x . q(\tilde{\Psi}^{\varepsilon, q}(u \oplus \langle \alpha_u \rangle(n)_{mnu}^x))).$$

Define $\tilde{\Phi}_{\varepsilon q} := \tilde{\Psi}^{\varepsilon, q}(\emptyset)$. Then it follows immediately from $\forall u B(u)$ and \prec'_\emptyset -induction that $\tilde{\Phi}_{\varepsilon q} = \alpha_\emptyset \text{ mr } \exists \alpha \forall n A(n)$ since $\forall n \alpha_\emptyset(n)_1 \text{ mr } A_n(\{(\alpha_\emptyset)_0\}(n), \alpha_\emptyset(n)_1)$. \square

5.2 Examples

We now show that essentially all of the solutions to the modified realizability interpretation choice principles given across e.g. [3, 6, 15, 26] appear as special cases of Theorem 5.3, given suitable instantiations of \triangleleft , \prec , m and the choice formula $A_n(s, x)$.

5.2.1 The Berardi-Bezem-Coquand functional ($\triangleleft = \prec_u = \emptyset$)

It is easy to see that the principle \emptyset -DC_{seq} is just countable choice, since ρ^{\emptyset_n} is just a singleton object $\{\bullet\}$ indexed by n , and so ρ^{\emptyset} is isomorphic to \mathbb{N} . Setting $A_n(\bullet, x) := B_n(x)$ we obtain AC as defined in Section 4. In fact, Theorem 5.3 completely reduces to Theorem 4.6 for the formula $A(n) := \exists x B_n(x)$ once we eliminate \triangleleft . Therefore as expected there is a direct correspondence between the realizers of the two theorems in this case.

For any function m satisfying $n \notin \text{dom}(u) \rightarrow mnu \notin \text{dom}(u)$, we can define a generalised version of the Berardi-Bezem-Coquand functional as $\text{BBC}_{(m)} := \Psi_{(\emptyset, \emptyset, m)}$, which satisfies the defining equation

$$\text{BBC}_{(m)}^{\varepsilon, q}(u) = q(u \oplus \lambda n . \varepsilon_{mnu, n}(\lambda x . \text{BBC}_{(m)}^{\varepsilon, q}(u_{mnu}^x))).$$

Regardless of the choice of m , this functional always gives a computational interpretation to countable choice. Now suppose that we move back into the more conventional setting of the double negation translation of countable choice, setting $R = \perp$ and assuming $B_n(x)$ is a negated formula, so that there exists a term h satisfying $\forall n, x (h \text{ mr } (\perp \rightarrow B_n(x)))$. Then if ϕ satisfies

$$\phi \text{ mr } \forall m ((\exists x B_m(x) \rightarrow \perp) \rightarrow \perp)$$

then $\varepsilon_{m, n}^{\phi}(p) :=_{\rho \times \sigma} \langle 0_{\rho}, h(\phi_m(p)) \rangle$ realizes the premise of \emptyset -DC_{seq}^{J₁*}, and thus defining $\text{BBC}_{(m), 1}^{\phi, q} := \text{BBC}_{(m)}^{\varepsilon^{\phi}, q}$ - which satisfies the equation

$$\text{BBC}_{(m), 1}^{\phi, q}(u) = q(u \oplus \lambda n . \langle 0, h(\phi_{mnu}(\lambda x . \text{BBC}_{(m), 1}^{\varepsilon, q}(u_{mnu}^x))) \rangle).$$

- we obtain a term which realizes the negative translation of AC. In particular, setting $mnu = n$ we obtain

$$\text{BBC}_2^{\phi, q}(u) = q(u \oplus \lambda n . \langle 0, h(\phi_n(\lambda x . \text{BBC}_2^{\varepsilon, q}(u_n^x))) \rangle)$$

which is just the BBC functional of [3] (more precisely, the variant of the Berardi-Bezem-Coquand functional for input with arbitrary domain considered in [5]).

5.2.2 Modified bar recursion ($\prec_u = \emptyset$)

Suppose that we retain the simplification $\prec_u = \emptyset$, but now allow \triangleleft to range over arbitrary decidable partial orders. Then we obtain a realizer which makes recursive calls over updates of its input, just like the BBC functional, but now m must not only satisfy conditions (i) and (ii) but also (5.3):

$$u \text{ is } \triangleleft\text{-closed} \rightarrow u \cup \{mnu\} \text{ is } \triangleleft\text{-closed}$$

In this case, the term $\text{MBR}_{(\triangleleft, m)}^{\varepsilon, q} := \Psi_{(\triangleleft, \emptyset, m)}^{\varepsilon, q}$, which has defining equation

$$\text{MBR}_{(\triangleleft, m)}^{\varepsilon, q}(u) = q(u \oplus \lambda n . \varepsilon_{\{\alpha\}(mnu), \{\alpha_0\}(n)}(\lambda x . \text{MBR}_{(\triangleleft, m)}^{\varepsilon, q}(u_{mnu}^x)))$$

forms a realizer for \triangleleft -DC_{seq}. This can be viewed as a generalisation of modified bar recursion as first defined in [6]. All existing variants of MBR occur when \triangleleft is the usual ordering on \mathbb{N} , but MBR is perfectly well-defined in more unusual cases. For instance, suppose that we have a bijective encoding $c: \mathbb{N} \rightarrow \mathbb{B}^*$, and that

$$m \triangleleft n := c(m) \text{ is a proper prefix of } c(n).$$

Then \triangleleft -closed partial functions are precisely partial functions whose domain is a binary tree, and in this case there are many valid choices for m , a canonical one being

$$mnu := n \text{ if } n \in \text{dom}(u) \text{ else } i \text{ where } c(i) \text{ is the least prefix of } c(n) \text{ not in } \text{dom}(u).$$

This variant of MBR yields an intuitive realizer for dependent choice over binary trees.

Now, suppose that we do indeed have $\triangleleft = \prec$. Then \prec -closed partial functions are either total or of the form \widehat{s} where s is a finite sequence. Observing that in order to evaluate $\text{MBR}(\emptyset)$ we can restrict ourselves to input with finite domain we can redefine our realizer in this case as $\text{MBR}_1^{\varepsilon, q}(s^{\rho^*}) := \text{MBR}_{(\prec, m)}^{\varepsilon, q}(\widehat{s})$, setting $mnu := \mu i \leq n (i \notin \text{dom}(u))$. Clearly such an m satisfies (5.3), and in particular $mn\widehat{s} = |s|$ for $n \geq |s|$. Therefore MBR_1 has defining equation

$$\text{MBR}_1^{\varepsilon, q}(s) = q(\widehat{s} \oplus \lambda n . \varepsilon_{s, \{\alpha_0\}}(n)(\lambda x . \text{MBR}_1^{\varepsilon, q}(s * x)))$$

This directly realizes $\prec\text{-DC}_{\text{seq}}$, which is isomorphic to

$$\forall s^{\rho^*} (\forall i < |s| A_i([s](i), s(i)) \rightarrow \exists x A_{|s|}(s, x)) \rightarrow \exists \alpha \forall n A_n([\alpha](n), \alpha(n)).$$

We can now easily rederive various concrete instances of MBR_1 found in the literature which arise from setting $R = \perp$ and instantiating $A_i(s, x)$ by the correct formula. First, note that we immediately derive AC from $\prec\text{-DC}_{\text{seq}}$ by setting $A_n(s, x) := B_n(x)$, and a corresponding realizer for AC by defining $\text{MBR}_2^{\phi, q} := \text{MBR}_1^{\varepsilon, q}$ for $\varepsilon_{s, r}^{\phi}(p) = \langle 0, h(\phi_{|s|}(p)) \rangle$, where ϕ and h are as in Section 5.2.1. This has defining equation

$$\text{MBR}_2^{\phi, q}(s) = q(\widehat{s} \oplus \lambda n . \langle 0, h(\phi_{|s|}(\lambda x . \text{MBR}_2^{\phi, q}(s * x))) \rangle)$$

which is exactly the realizer of countable choice constructed in [6]. Note that the same realizer could have been constructed from $\text{BBC}_{(m), 1}$ for suitable m .

Now suppose that $A_0(s, x) := B_0(x_0, x)$ and $A_n(s, x) := B_n(s_{|s|-1}, x)$ for $n > 0$. Then $\prec\text{-DC}_{\text{seq}}$ immediately implies the following, standard formulation of dependent choice:

$$\text{DC} : \forall n, y \exists x B_n(y, x) \rightarrow \forall x_0 \exists \alpha (\alpha(0) = x_0 \wedge \forall n B_n(\alpha(n), \alpha(n+1))).$$

The challenge for realizing DC is as follows: we must construct a realizer of \perp , given realizers

$$\begin{aligned} \phi^{\mathbb{N} \rightarrow \rho \rightarrow (\rho \times \sigma) \rightarrow \tau} \rightarrow \tau \quad mr \quad \forall n, y ((\exists x B_n(y, x) \rightarrow \perp) \rightarrow \perp) \\ Y^{(\rho \times \sigma)^{\mathbb{N}} \rightarrow \tau} \quad mr \quad \exists \alpha (\alpha(0) = x_0 \wedge \forall n B_n(\alpha(n), \alpha(n+1))) \rightarrow \perp \end{aligned}$$

and in addition assuming that B is negated, thus guaranteeing the existence of a realizer h of ex-falso quodlibet satisfying $\forall n, y, x h \text{ } mr (\perp \rightarrow B_n(y, x))$. But in this case, we can easily define realizers of the premise of $\prec\text{-DC}_{\text{seq}}$ as

$$\begin{aligned} \varepsilon_{s, r}^{\phi}(p) &:= \langle 0, h(\phi_{|s|, (x_0 * s_0)_{|s|}}(p)) \rangle \\ q^Y(\alpha) &:= Y_{x_0}(\alpha) := Y(\langle x_0 * \alpha_0, \alpha_1 \rangle) \end{aligned}$$

and defining $\text{MBR}_3^{\phi, Y} := \text{MBR}_1^{\varepsilon, q^Y}$ yields a realizer for DC satisfying

$$\text{MBR}_3^{\phi, Y}(s) = Y_{x_0}(\widehat{s} \oplus \lambda n . \langle 0, h(\phi_{|s|, (x_0 * s_0)_{|s|}}(\lambda x . \text{MBR}_3^{\phi, Y}(s * x))) \rangle)$$

which is this time exactly the realizer of DC given in [6]. In an entirely analogous way, the bar recursive solution to the sequential variant of dependent choice considered in [26]:

$$B(\langle \rangle) \rightarrow \forall s^{\rho^*} (B(s) \rightarrow \exists x B(s * x)) \rightarrow \exists \alpha \forall n B([\alpha](n)) \tag{9}$$

can be defined in terms of MBR_1 , since (9) is easily implied by $\prec\text{-DC}_{\text{seq}}$ for $A_n(s, x) := B(s * x)$. Given realizers

$$\begin{aligned} a_0^{\rho} \quad mr \quad B(\langle \rangle) \\ \phi^{\rho^* \rightarrow \sigma \rightarrow (\rho \times \sigma \rightarrow \tau) \rightarrow \tau} \quad mr \quad \forall s (B(s) \rightarrow (\exists x B(s * x) \rightarrow \perp) \rightarrow \perp) \\ Y^{(\rho \times \sigma)^{\mathbb{N}} \rightarrow \tau} \quad mr \quad \exists \alpha \forall n B([\alpha](n)) \rightarrow \perp \end{aligned}$$

and assuming the existence of a realizer $\forall s h \text{ } mr (R \rightarrow B(s))$, it is easy to see that

$$\begin{aligned} \varepsilon_{s, r}^{a_0, \phi}(p) &:= \langle 0, h(\phi_{s_0, (a_0 * s)_{|s|}}(p)) \rangle \\ q^{a_0, Y}(\alpha) &:= Y_{a_0}(\alpha) := Y(\langle \alpha_0, G_0 * \alpha_1 \rangle) \end{aligned}$$

realize the premise of $\prec\text{-DC}_{\text{seq}}$, and therefore the realizer we obtain is $\text{MBR}_4^{a_0, \phi, Y} := \text{MBR}_1^{\varepsilon_{s, r}^{a_0, \phi}, q^{a_0, Y}}$ which satisfies

$$\text{MBR}_4^{a_0, \phi, Y}(s) = Y_{a_0}(\widehat{s} \oplus \lambda n . \langle 0, h(\phi_{s_0, (a_0 * s)_{|s|}})(\lambda x . \text{MBR}_4^{a_0, \phi, Y}(s * x)) \rangle)$$

which is exactly the term used to interpret dependent choice in [26] (and also extract an algorithm from Higman's lemma in [25]).

5.2.3 Products of selection functions ($mnu = n$, \triangleleft total)

We finally consider the case in which the functional m is defined to be the projection function $mnu = n$. Clearly, the conditions (i)-(iii) are satisfied whenever $\prec_u = \triangleleft$, in which case the realizer simplifies to

$$\Psi^{\varepsilon, q}(u) = q(u \oplus \lambda n . \underbrace{\varepsilon_{\{\alpha\}(n)}(\lambda x . \Psi^{\varepsilon, q}(u \oplus \{\alpha\}(n)_n^x))}_{\alpha_u})$$

where we abbreviate $\varepsilon_{\{\alpha\}(n)} = \varepsilon_{\{\alpha\}(n), \{\alpha_0\}(n)}$ and $\{\alpha\}(n)$ is now treated as a partial function. Now, suppose that \triangleleft is a *total* order, and therefore constitutes an encoding of some countable ordinal ξ . It is always the case that we can evaluate $\Psi(\emptyset)$ by restricting the input to being \triangleleft -closed. However, when \triangleleft is total then the set of (domain-theoretically) non-total \triangleleft -closed partial sequences is isomorphic to ρ^{\triangleleft} , since if u is \triangleleft -closed then $\text{dom}(u) = \{k \mid k \triangleleft n\}$ where n is the least element of the set of undefined elements of u . Let us suppose that \triangleleft comes equipped with a computable ordinal successor function $\text{succ}: \mathbb{N} \rightarrow \mathbb{N}$ i.e. for each number n , $\text{succ}(n)$ is the least number greater than n with respect to \triangleleft . Define $\text{PS}_{(\triangleleft)}^{\varepsilon, q}: \rho^{\triangleleft} \rightarrow \rho^{\mathbb{N}}$ by $\text{PS}_{(\triangleleft)}^{\varepsilon, q}(s) := \tilde{\Psi}^{\varepsilon, q}(\widehat{s})$ (where $\tilde{\Psi}$ is the term defined in Corollary 5.6). Then PS satisfies the equation

$$\text{PS}_{(\triangleleft)}^{\varepsilon, q}(s) = \widehat{s} \oplus \lambda n . \varepsilon_{\{\alpha\}(n)}(\lambda x . q(\text{PS}_{(\triangleleft)}^{\varepsilon, q}(\{\alpha\}(n) * x))).$$

where $\{\alpha\}(n) * x: \rho^{\triangleleft_{\text{succ}(n)}}$ is defined by

$$(\{\alpha\}(n) * x)(m) := \begin{cases} \alpha(m) & \text{if } m \triangleleft n \\ x & \text{if } m = n. \end{cases}$$

This functional is a generalisation of the (implicitly well-founded) product of selection functions of Escardó and Oliva to arbitrary recursive ordinals, and is not only a realizer $\triangleleft\text{-DC}_{\text{seq}}^{J_2^*}$, but a direct realizer of the following transfinite J -shift principle:

$$\begin{cases} \forall s (\forall i \triangleleft |s| A_i(\{s\}(i), s(i)) \rightarrow (\exists x A_{|s|}(s, x) \rightarrow R) \rightarrow \exists x A_{|s|}(s, x)) \\ \rightarrow (\exists \alpha \forall n A_n(\{\alpha\}(n), \alpha(n)) \rightarrow R) \rightarrow \exists \alpha \forall n A_n(\{\alpha\}(n), \alpha(n)). \end{cases}$$

For the particular case that \triangleleft is the normal ordering on \mathbb{N} and $A_n(s, x) := B_n(s * x)$ this becomes

$$\begin{cases} \forall s^{\rho^*} (\forall i < |s| B_i([s](i+1)) \rightarrow (\exists x B_{|s|}(s * x) \rightarrow R) \rightarrow \exists x B_{|s|}(s * x)) \\ \rightarrow (\exists \alpha \forall n B_n([\alpha](n+1)) \rightarrow R) \rightarrow \exists \alpha \forall n B_n([\alpha](n+1)). \end{cases}$$

which is precisely the dependent J -shift of [15], and our realizer becomes

$$\text{PS}_{(\triangleleft)}^{\varepsilon, q}(s) = \widehat{s} \oplus \lambda n . \varepsilon_{[\alpha](n)}(\lambda x . q(\text{PS}_{(\triangleleft)}^{\varepsilon, q}([\alpha](n) * x)))$$

which is isomorphic to the dependent product of selection functions of [15]. Setting $A_n(s, x) := B_n(x)$ as in Section 5.2.2 we see that the dependent J -shift implies the non-dependent J -shift:

$$\forall n (\exists x B_n(x) \rightarrow R) \rightarrow \exists x B_n(x) \rightarrow (\exists \alpha \forall n B_n(\alpha(n)) \rightarrow R) \rightarrow \exists \alpha \forall n B_n(\alpha(n))$$

and our realizer can be simplified to

$$\text{PS}_{(\triangleleft)}^{\varepsilon, q}(s) = \widehat{s} \oplus \lambda n . \varepsilon_n(\lambda x . q(\text{PS}_{(\triangleleft)}^{\varepsilon, q}([\alpha](n) * x)))$$

which is now isomorphic to the non-dependent product of selection functions of [15].

Summary

We have demonstrated that the many variants of bar recursion used to interpret choice principles are just instances of the same basic combinatorial idea. Theorem 5.3 gives us a completely uniform framework in which to understand the variety of different realizers currently in the literature, and moreover in each case we are able to provide generalisations of these realizers to more complex orderings on \mathbb{N} . We summarise all this in the table below (here $m_1nu := \mu i \leq n (i \notin \text{dom}(u))$ and $m_2nu = n$).

m	\prec_u	\triangleleft	$A_n(s, x)$	realizer
m_1	\emptyset	\emptyset	$B_n(x)$	Berardi-Bezem-Coquand functional [3]
m_2	\emptyset	$\emptyset / <$	$B_n(x)$	simple modified bar recursion [6]
m_2	\emptyset	$<$	$B_n(s _{s -1}, x)$	dependent modified bar recursion [6]
m_2	\emptyset	$<$	$B(s * x)$	dependent modified bar recursion [26]
m_1	$<$	$\emptyset / <$	$B_n(x)$	simple product of selection functions [12, 15]
m_1	$<$	$<$	$B_n(s * x)$	dependent product of selection functions [12, 15]

However, these examples constitute only an extremely limited range of possibilities for $\Psi_{(m, \triangleleft, <)}$ based on very simple instantiations of its parameters. Theorem 5.3 gives us far more than just a unifying perspective for existing realizers: it gives us a very general recipe for devising new realizers for choice principles that can be tailored to the situation at hand. Thus, rather than simply giving a fixed interpretation of choice and extracting programs relative to this realizer, we have an additional level of flexibility which should allow us to extract much more efficient and meaningful programs.

6 Understanding the parametrised realizer

Having completed the main theoretic work of this paper, the purpose of this section is to give a somewhat informal graphical representation of the structure of our realizer and the proof of Theorem 5.3. In doing so we suggest a potential semantic interpretation of the realizer.

Much work has been done in the last few decades on providing a computational interpretation of classical logic that can be understood on an intuitive level terms of *learning* - the basic idea being that realizers of classical principles typically carry out some kind of ‘learning procedure’ in order to construct an approximation to that principle.

In [3] a connection is suggested between realizers of negative translated formulas and winning strategies related to the Novikoff interpretation of classical formulas [11, 20]. In particular, an illuminating semantic interpretation of what we have called the Berardi-Bezem-Coquand functional is given, in which the functional represents a strategy for building an approximation of a choice functional which wins against any continuous opponent. We attempt to extend this interpretation to our generalised realizer of choice, and argue that the our parameters can be giving a clear meaning in this context.

Let us first very briefly recall the basic ideas of [3, 11] (the reader is referred to these papers for a proper treatment). In the Novikoff calculus, the formula $\exists x \forall y A(x, y)$ is mapped to the propositional formula

$$\bigvee_x \bigwedge_y A(x, y).$$

The truth of such a formula is debated by two players \exists loise and \forall belard, who support truth and falsity respectively. First, \exists loise selects some value x_0 for which she claims that $\bigwedge_y A(x_0, y)$ is true, and \forall belard follows this by choosing some y_0 in an attempt to falsify $A(x_0, y_0)$. The formula as a whole is intuitionistically valid iff \exists loise has a winning strategy regardless of any choices made by \forall belard.

Now, it is clear that such a correspondence does not work in the case of classical logic, since there are Σ_2 -formulas classically true but for which there is no effective strategy for $\bigvee_x \bigwedge_y A(x, y)$. In this case, validity of $\bigvee_x \bigwedge_y A(x, y)$ is interpreted as the existence of some x_0 such that

$$A(x_0, y') \vee \bigvee_x \bigwedge_y A(x, y)$$

is valid for all y' . The idea here is that \exists loise picks a potential witness x_0 , which is followed by an attempt at a counterexample y_0 from \forall belard, and the game becomes $A(x_0, y_0) \vee \bigvee_x \bigwedge_y A(x, y)$. In other words, either $A(x_0, y_0)$ is true, in which case \exists loise wins, or it is false, and \exists loise can backtrack and start again, this time using falsity of $A(x_0, y_0)$ as constructive information. In this way, \exists loise is allowed to ‘learn’ from \forall belard’s choices. Moreover, as demonstrated in [3], this notion of learning and backtracking is captured by the recursive functionals which realize the negative translation of the classical formula.

Let us now take this basic idea and consider how the truth of countable choice can be interpreted as a dialogue between \exists loise and \forall belard in which \exists loise eventually wins. Countable choice can be written as the disjunction

$$\exists n \forall x \neg A_n(x) \vee \exists f \forall n A_n(f(n)).$$

We give this formula a rough interpretation along the lines of [3] as follows. \exists loise begins by attempting to realize the conclusion of AC with some default function $f_0 = \lambda n.0$ whose ‘domain’ of genuine constructive information is empty, and \forall belard responds by selecting some point n_0 such that this attempt fails i.e. we cannot provide a realizer for $A_{n_0}(0)$. \exists loise responds by now attempting to falsify the premise at some m_0 , to which \forall belard responds with a point x_0 and a realizer for $A_{m_0}(x_0)$.

\exists loise has now been given some constructive information by \forall belard, so she starts the whole process again, this time with the function $f_1 := f_0[m_0 \mapsto x_0]$, which now has a domain of $\{m_0\}$. This time, \forall belard picks a point n_1 at which f_1 fails. Either he picks $n_1 = m_0$ in which case he loses since by his own admission $A_{m_0}(x_0)$ is true (see [3] for a formal translation of this logic into Novikoff strategies), or he chooses $n_1 \notin \{m_0\}$. Then \exists loise responds with some $m_1 \notin \{m_0\}$ which falsifies the premise of AC, and again \forall belard responds with some x_1 and a realizer for $A_{m_1}(x_1)$. \exists loise now updates her approximation again to some f_2 which includes this information, and continues as before. Roughly speaking, such a strategy should eventually result in success for \exists loise whenever \forall belard is ‘continuous’, because he will eventually be forced to pick n_i in the domain of f_i .

We have been vague as to two details in this strategy: firstly how \exists loise decides on which m_i to choose in light of \forall belard’s original choice n_i , and secondly in how she chooses to update her approximation each time. In the first instance it is clear that if \forall belard has a ‘good’ choice for n_i which is not in the domain f_i then \exists loise must also respond with m_i not in the domain of f_i if she has any chance of falsifying the premise of AC. Then, when it comes to updating f_i with the information (m_i, x_i) , she could either just add this directly to f_i and define $f_{i+1} := f_i[m_i \mapsto x_i]$, or she could potentially erase some of the existing elements in the domain of f_i .

The most natural choice is of course to pick $m_i = n_i$ each time and update directly without any erasing, and this is precisely the strategy given in [3]. However, in the case of *dependent choice* where the elements of the choice sequences are related in some way, it is crucial for \exists loise to update her approximation in a manner which is coherent with the underlying dependency required for the choice sequence.

Let us now move on to our realizer for the negative translation of countable dependent choice (in the form of the \triangleleft -DC $_{\text{seq}}^{J_1^*}$ -shift), and show how the parameters of the realizer correspond to the choices in \exists loise’s strategy we have just discussed. In Figure 1 below we give a (very informal) diagrammatic representation of the proof of Theorem 5.3. We retain all our notation, in particular we write $\bar{A}(u)$ to denote that u is a partial realizer. For simplicity we now work in a concrete realizability setting with $R = \perp$ and $\varepsilon_{s,r}(p) = \langle 0, h(\phi_s(p)) \rangle$ where

$$\phi \text{ } mr \forall s((\exists x A_{|s|}(s, x) \rightarrow \perp) \rightarrow \perp).$$

and $\forall n, r, x(h \text{ } mr (\perp \rightarrow A_n(r, x)))$, so in particular ε realizes the premise of \triangleleft -DC $_{\text{seq}}^{J_1^*}$.

Let us now run through the diagram, step by step. Roughly speaking, each box represents a stage in a game between \exists loise and \forall belard, and an arrow represents a reverse implication in the proof of Theorem 5.3. We start at A with the assumption that \exists loise has already computed a partial realizer u that is correct wherever it’s defined. At step B , \exists loise plays an approximation $(\alpha_u)_0 := u_0 \oplus 0$ to a choice sequence with a corresponding sequence $(\alpha_u)_1$ of realizers, and in response, \forall belard selects some n and challenges \exists loise to realise $A_n(\{(\alpha_u)_0\}(n), (\alpha_u)_0)$. If $n \in \text{dom}(u)$ then \exists loise wins, so we assume the contrary and move on to step C . \exists loise responds to the challenge in the next step D by claiming that the premise of choice principle is false at point mnu . \forall belard is now forced to produce x such that $x_1 \text{ } mr A_{mnu}(\{(\alpha_u)_0\}(mnu), x_0)$ - if he fails then \exists loise wins, and if he succeeds then \exists loise is given constructive information and takes this as an assumption.

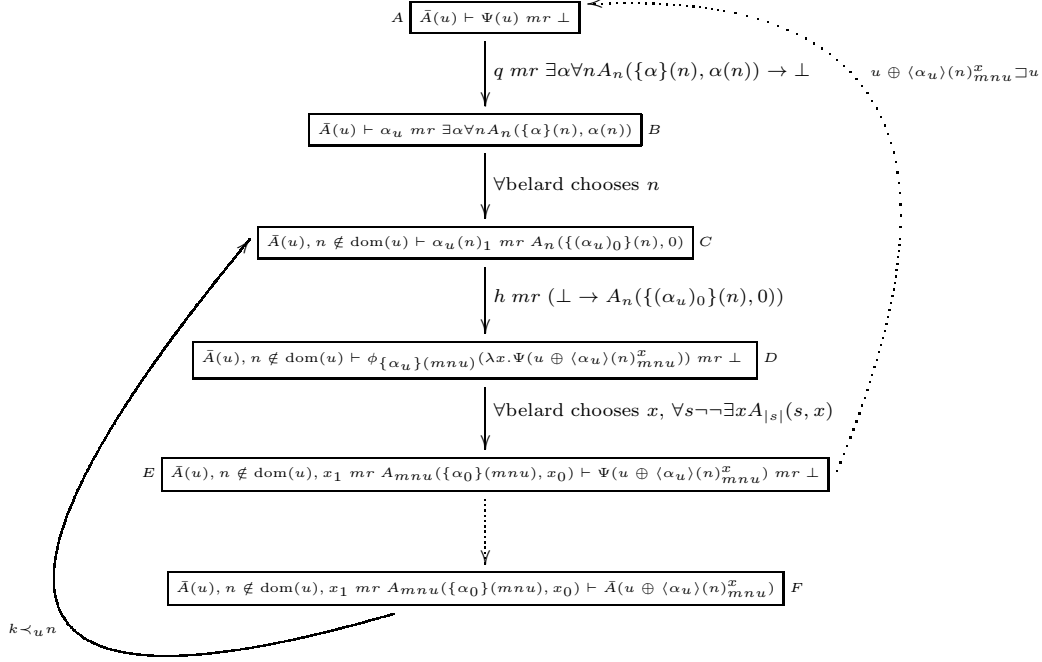


Figure 1: Diagrammatic illustration of Theorem 5.3

We now come to the subtle part: the manner in which \exists loise updates $(\alpha_u)_0$ to reflect this new information and repeat the loop. From a proof theoretic perspective, E is implied by A and F by the cut rule. We can interpret this semantically as follows: \exists loise forms the updated function $(\alpha_u)_0[mnu \mapsto x_0]$, but states that if in future \forall belard queries this realizer for any $k \prec_u n$ with $k \notin \text{dom}(u)$, she will ignore any subsequent information received and revert to stage C as if \forall belard had chosen k instead of n . This is reflected in the definition of Ψ , since it makes a recursive call on the partial function $u \oplus \langle \alpha_u \rangle(n)_{mnu}^x$, but the fact that this is a partial realizer for $k \prec_u n$ relies on nested recursive calls of the form $\Psi(u \oplus \langle \alpha \rangle(k)_{mku}^y)$, which in particular forgets the value of x at mnu .

Therefore at stage E there are two possibilities. Either at some point in the future \forall belard does query $k \prec_u n$ for $k \notin \text{dom}(u)$, in which case all subsequent information is deemed irrelevant and we make an auxiliary \prec_u -recursive loop back to C , or \forall belard never queries $k \prec_u n$, in which case we can treat $u \oplus \langle \alpha_u \rangle(n)_{mnu}^x$ as a partial realizer and make a backward recursive loop back to stage A . In this way, the game corresponding to dependent choice can be seen as a path through Figure 1. By the combination of \prec_u -induction and backward induction, using the fact that \prec_u is well-founded and \forall belard's choice of n at step B is based on a continuous strategy, the proof of Theorem 5.3 essentially says that there is no infinite path starting from \emptyset , and therefore \exists loise has a winning strategy in the game as a whole.

Of course, we have not made our proposed link between Theorem 5.3 and the world of Novikoff games precise! But at the very least we hope to have provided some insight into how our somewhat syntactic construction in the previous sections could be viewed from an intuitive, semantic perspective.

Let us now think about our concrete examples. For the Berardi-Bezem-Coquand functional we have $\langle \alpha_u \rangle(n) = \emptyset$ and so the 'forgetful' subloop is completely avoided. In this case \exists loise simply responds to \forall belard's choice of n directly, and so when \forall belard plays $(\langle n_i, x_i \rangle) = \langle 1, x_0 \rangle, \langle 4, x_1 \rangle, \langle 3, x_2 \rangle, \dots$ we get the following sequence of updates

$$[0, 0, 0, 0, 0] \mapsto [0, x_0, 0, 0, 0] \mapsto [0, x_0, 0, 0, x_1] \mapsto [0, x_0, 0, x_2, x_1] \mapsto \dots$$

This semantic interpretation of the BBC-functional is of course completely analogous to the one given in [3]. For modified bar recursion, whenever \forall belard makes a sensible choice $n \notin \text{dom}(u)$, \exists loise switches instead the least element not already in the domain. Again we avoid the forgetful subloop, but this time \exists loise's updates are done in sequence - the same three choices from \forall belard results in the following

response:

$$[0, 0, 0, 0, 0] \mapsto [x_0, 0, 0, 0, 0] \mapsto [x_0, x_1, 0, 0, 0] \mapsto [x_0, x_1, x_2, 0, 0] \mapsto \dots$$

Finally, for the product of selection functions, \exists loise never changes \forall belard's choice of n , but now assumes a policy of forgetting everthing above the point being updated. This time the result is

$$[0, 0, 0, 0, 0] \mapsto [0, x_0, 0, 0, 0] \mapsto [0, x_0, 0, 0, x_1] \mapsto [0, x_0, 0, x_2, 0] \mapsto \dots$$

the point x_1 being forgotten as a result of \forall belard choosing $3 < 4$ as his third move.

The strategy related to the BBC-functional fails for dependent choice because it is assumed that \forall belard always picks x_i such that $A_m([u \oplus 0](m), x_i)$ holds, and so e.g. we have $A_4([0, x_0, 0, 0], x_1)$. However, if ever in the future he gives \exists loise some new information x_{i+j} for point $k < m$ in the sequence, then x_i is no longer valid - for instance we would require $A_4([0, x_0, 0, x_2], x_1)$ to hold, which is not necessarily true.

Modified bar recursion and the product of selection functions represent two methods of overcoming this. For the former, \exists loise ensures that \forall belard always gives her constructive information in sequence, while for the latter she is happy to take information for any point \forall belard chooses, but whenever updating she erases everthing above, relying on future moves to regain this information.

7 Concluding remarks

There are several directions in which the work presented here could be developed. We have already demonstrated that Theorem 5.3 provides a uniform soundness proof through which most of the existing variants of backward recursion used to interpret dependent choice can be derived. The most obvious next step would be to explore the use of new variants of bar recursion which arise from giving more interesting values to the parameters of $\Psi_{(m, \prec, \triangleleft)}$.

Take, for example, the following classical statement:

For any function $f: \mathbb{B}^* \rightarrow \mathbb{N}$, there exists a function $g: \mathbb{B}^* \rightarrow \mathbb{N}$ such that for any branch $s: \mathbb{B}^*$ we have $f(g(s)) \leq f(t)$ whenever t has s as a prefix.

This is a direct consequence of the axiom of countable choice, and so any instantiations of (m, \prec) satisfying the conditions of Theorem 4.6 will yield a realizer capable of building an approximation to the choice function g . However, it would seem most sensible to choose the parameters so that the recursion is carried out over the natural tree structure $s \triangleleft t$ iff s is a prefix of t (relative to some encoding of \mathbb{B}^* into \mathbb{N}) - using, for example, the variant of modified bar recursion sketched out the beginning of Section 5.2.2. Then if an approximation to g is defined at s then this information could be used to extend the approximation to extensions t of s .

It would be interesting to examine in general how such realizers, tailored to the situation at hand, compare to those built from the existing forms of bar recursion. One would expect an advantage in terms of both algorithmic efficiency and the syntactic expressiveness of the extracted program.

Another interesting application of our parametrised form of bar recursion would be to extend the work of Escardó and Oliva on Nash-equilibria of unbounded games to the transfinite case. In [14, 15] it is shown that the infinite products of selection functions corresponding to the functional $\text{PS}_{(\prec)}$ defined in Section 5.2.3 computes optimal strategies in infinite sequential games over the natural numbers. However, $\text{PS}_{(\triangleleft)}$ is well-defined for any computable well ordering on \mathbb{N} , and in particular it is not too difficult to show that in these cases Spector's equations:

$$\begin{aligned} \alpha(n) &= \varepsilon_{\{\alpha\}(n)}(p_n) \\ p_n(\varepsilon_{\{\alpha\}(n)}(p_n)) &= q(\alpha) \end{aligned}$$

can be solved in ε and q for arbitrary \triangleleft by setting $\alpha = \text{PS}^{\varepsilon, q}(\emptyset)$ and $p_n(x) := q(\text{PS}^{\varepsilon, q}(\{\alpha\}(n) \star x))$, entirely analogously to the normal ordering on \mathbb{N} . However, it would be useful to formalise this properly and to investigate whether there are any interesting applications of higher-type *transfinite* games.

Finally, to the author's knowledge all known realizability interpretations of choice are restricted to countable or dependent choice principles, and it is not known how to extend these to choice over function spaces, for example:

$$\forall f^{\mathbb{N} \rightarrow \mathbb{N}} \exists x^\rho B_f(x) \rightarrow \exists F^{(\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \rho} \forall f B_f(F(f)).$$

One possibility would be to build an approximation to F over some countable basis $(c_i)_{i \in \mathbb{N}}$ for the space $\mathbb{N} \rightarrow \mathbb{N}$. However, in this case updates to the approximation must be made in a coherent way, since the values of two distinct elements c_1 and c_2 must be compatible with their intersection. Therefore the ideas behind our parametrised realizer could be helpful here, the aim being to assign an ordering to the basis and make sure that the updates respect this ordering. However, in the author's opinion it is likely that some additional ingenuity would be required here, as the challenge posed by giving a computational interpretation non-countable choice seems to be a significant one.

Funding. This work was supported by a LabEx CARMIN postdoctoral research fellowship, and also the Austrian Science Fund (FWF) project ‘‘Automated Complexity Analysis via Transformations’’ (project number P25781).

A Appendix

Proof of Proposition 5.2. We define $\Psi_{\prec'_u, \prec_u, m}^{\varepsilon, q}(u) := \text{BkR}^{\psi^{\varepsilon, q}}(u)$ where

$$\psi_u^{\varepsilon, q}(f^{\mathbb{N} \times \overline{(\rho \times \sigma)^{\mathbb{N}}} \rightarrow \tau}) :=_{\tau} q(\alpha_u^{\varepsilon, f})$$

where $\alpha_u^{\varepsilon, f} : (\rho \times \sigma)^{\mathbb{N}}$ is recursively defined over \prec'_u by

$$\alpha_u^{\varepsilon, f}(n) := \begin{cases} u(n) & \text{if } n \in \text{dom}(u) \\ \varepsilon_{\{\alpha\}(mnu), \{\alpha_0\}(n)}(\lambda x . f(mnu, u \oplus \langle \alpha_u \rangle(n)_{mnu}^x)) & \text{otherwise} \end{cases}$$

where $\langle \alpha_u \rangle(n) := \lambda k. (\alpha_u^{\varepsilon, f}(k) \text{ if } k \prec_u n)$. By condition (i) of the proposition, $\alpha_u^{\varepsilon, f}$ is well-founded. Now, unwinding definitions, we obtain

$$\begin{aligned} \Psi^{\varepsilon, q}(u) &= \psi_u(\lambda n, v . \underbrace{\Psi^{\varepsilon, q}(u \oplus v)}_f) \text{ if } n \in \text{dom}(v) \setminus \text{dom}(u) \\ &= q(\alpha_u) \\ &= q(u \oplus \lambda n . \varepsilon_{\{\alpha\}(mnu), \{\alpha_0\}(n)}(\lambda x . f(mnu, u \oplus \langle \alpha_u \rangle(n)_{mnu}^x))) \\ &\stackrel{(*)}{=} q(u \oplus \lambda n . \varepsilon_{\{\alpha\}(mnu), \{\alpha_0\}(n)}(\lambda x . \Psi(u \oplus \langle \alpha_u \rangle(n)_{mnu}^x))) \end{aligned}$$

where step (*) follows from

$$\begin{aligned} f(mnu, u \oplus \langle \alpha_u \rangle(n)_{mnu}^x) &= \Psi(u \oplus (u \oplus \langle \alpha_u \rangle(n)_{mnu}^x)) \text{ if } mnu \in \text{dom}(u \oplus \langle \alpha_u \rangle(n)_{mnu}^x) \setminus \text{dom}(u) \\ &= \Psi(u \oplus \langle \alpha_u \rangle(n)_{mnu}^x) \text{ if } mnu \notin \text{dom}(u) \\ &= \Psi(u \oplus \langle \alpha_u \rangle(n)_{mnu}^x) \end{aligned}$$

the last step following from condition (ii). □

References

- [1] F. Aschieri. *Learning, Realizability and Games in Classical Arithmetic*. PhD thesis, Universita degli Studi di Torino and Queen Mary, University of London, 2011.
- [2] J. Avigad. Update procedures and the 1-consistency of arithmetic. *Mathematical Logic Quarterly*, 48(1):3–13, 2002.
- [3] 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.
- [4] 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.

- [5] 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.
- [6] U. Berger and P. Oliva. Modified bar recursion and classical dependent choice. *Lecture Notes in Logic*, 20:89–107, 2005.
- [7] U. Berger and P. Oliva. Modified bar recursion. *Mathematical Structures in Computer Science*, 16(2):163–183, 2006.
- [8] U. Berger and H. Schwichtenberg. Program extraction from classical proofs. In D. Leivant, editor, *Logic and Computational Complexity workshop (LCC'94)*, volume 960 of *Lecture Notes in Computer Science*, pages 77–97. 1995.
- [9] M. Bezem. Strongly majorizable functionals of finite type: A model for bar recursion containing discontinuous functionals. *Journal of Symbolic Logic*, 50:652–660, 1985.
- [10] V. Blot and C. Riba. On bar recursion and choice in a classical setting. In *Proceedings of APLAS 2013*, volume 8301 of *Lecture Notes in Computer Science*, pages 349–364, 2013.
- [11] T. Coquand. An analysis of Ramsey’s theorem. *Information and Computation*, 110(2):297–304, 1994.
- [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. The Peirce translation. *Annals of Pure and Applied Logic*, 163(6):681–692, 2012.
- [16] S. C. Kleene. Countable functionals. In A. Heyting, editor, *Constructivity in Mathematics*, pages 81–100. North-Holland, Amsterdam, 1959.
- [17] U. Kohlenbach. *Applied Proof Theory: Proof Interpretations and their Use in Mathematics*. Monographs in Mathematics. Springer, 2008.
- [18] 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.
- [19] C. St. J. A. Nash-Williams. On well-quasi-ordering finite trees. *Proceedings of the Cambridge Philosophical Society*, 59:833–835, 1963.
- [20] P.S. Novikoff. On the consistency of certain logical calculi. *Matematicheskij Sbornik*, 12(54):230–260, 1943.
- [21] P. Oliva and T. Powell. On Spector’s bar recursion. *Mathematical Logic Quarterly*, 58:356–365, 2012.
- [22] P. Oliva and T. Powell. Bar recursion over finite partial functions. 2014. Preprint, available from <http://arxiv.org/abs/1410.6361>.
- [23] P. Oliva and T. Powell. A constructive interpretation of Ramsey’s theorem via the product of selection functions. *To appear: Mathematical Structures in Computer Science*, 2015. Preprint available at <http://arxiv.org/abs/1204.5631>.
- [24] J.-C. Raoult. Proving open properties by induction. *Information Processing Letters*, 29:19–23, 1988.

- [25] M. Seisenberger. *On the Constructive Content of Proofs*. PhD thesis, Ludwig Maximilians Universität München, 2003.
- [26] M. Seisenberger. Programs from proofs using classical dependent choice. *Annals of Pure and Applied Logic*, 153(1-3):97–110, 2008.
- [27] 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.
- [28] T. Streicher. A classical realizability model arising from a stable model of untyped lambda calculus. Preprint, available from <http://arxiv.org/abs/1407.1547>, 2014.
- [29] A. S. Troelstra. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*, volume 344 of *Lecture Notes in Mathematics*. Springer, Berlin, 1973.