Annals of Pure and Applied Logic 163 (2012) 681–692

Contents lists available at SciVerse ScienceDirect

Annals of Pure and Applied Logic

journal homepage: www.elsevier.com/locate/apal

The Peirce translation

Martín Escardó [a], Paulo Oliva [b,*]

[a] University of Birmingham, UK
[b] Queen Mary University of London, UK

ARTICLE INFO

Article history:
Available online 23 November 2011

MSC:
03F03
03F07
03F10
18C20

Keywords:
The Peirce law
Negative translation
Countable choice
Dependent choice

ABSTRACT

We develop applications of selection functions to proof theory and computational extraction of witnesses from proofs in classical analysis. The main novelty is a translation of minimal logic plus Peirce law into minimal logic, which we refer to as *Peirce translation*, as it eliminates uses of Peirce law. When combined with modified realizability, this translation applies to full classical analysis, i.e. Peano arithmetic in the language of finite types extended with countable choice and dependent choice. A fundamental step in the interpretation is the realizability of a strengthening of the double-negation shift via the iterated product of selection functions. In a separate paper, we have shown that such a product of selection functions is equivalent, over system T, to modified bar recursion.

1. Introduction

Negative translations, also known as double negation translations, underpin virtually all computational interpretations of classical logic, arithmetic and analysis. First introduced as a way to reduce the consistency of classical arithmetic to that of intuitionistic arithmetic, these translations have proven to be useful also in computer science [12], set theory [1], arithmetic, and analysis [16].

Most negative translations are based on the so-called *continuation monad*, which associates each type of A with a new type

$$KA \equiv (A \to R) \to R.$$

When $R = \bot$, it corresponds to the double negation $\neg\neg A$ of A. In this paper, we consider a different translation based on the *Peirce monad*

$$JA \equiv (A \to R) \to A.$$

We call this the Peirce monad because the algebras of J are formulas satisfying Peirce law $JA \to A$. We have shown in [9] that the construction J over any cartesian closed category gives rise to a strong monad, with a monad morphism $\varepsilon \in JA \mapsto \phi \in KA$ from J to K as

$$\phi p = p(\varepsilon(p)). \tag{1}$$

Both J and K are strong monads, in the sense that we have morphisms

$$A \times TB \to T(A \times B),$$

---
* Corresponding author.
E-mail addresses: m.escardo@cs.bham.ac.uk (M. Escardó), paulo.oliva@eecs.qmul.ac.uk (P. Oliva).

0168-0072/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.apal.2011.11.002

for $T \in \{J, K\}$, satisfying certain equations. As a consequence of strength, we also have a product operation

$$TA \times TB \to T(A \times B).$$

In the previous work [9,6], we investigated the monad J from a general perspective, and showed that the product operation corresponding to the monad J can be seen as computing optimal strategies for a general definition of sequential games (cf. [10]). We have called elements $\varepsilon \in JA$ *selection functions* for the type A, as these can be viewed as selecting an element $\varepsilon p \in A$ for any given mapping $p \in A \to R$. In the concrete case when R is the set of Booleans $\mathbb{B}$, if ε always selects $x = \varepsilon p$ such that $p(x)$ holds, whenever that is possible, this corresponds to Hilbert's ε-operator in his ε-calculus. Moreover, as in the ε-calculus, one can define the existential quantifier from the ε-terms, we can also view elements of KA as *quantifiers*. Eq. (1) says that any selection function defines a quantifier.

In [6], the first author considered the particular case where the object A is a domain, and the object R is the domain of Boolean values. The particular quantifier ϕ studied was the bounded existential quantifier $\exists_S$ for a subset S of A, with the requirement that $\varepsilon(p)$ be an element of S such that if $p(s)$ holds for some $s \in S$, then $p(\varepsilon(p))$ holds, i.e. Eq. (1), for all $p \in A \to R$. The set $S \subseteq A$ is called *exhaustible* if the quantifier $\phi = \exists_S$ is computable, and *searchable* if additionally there is a computable functional $\varepsilon \in JA$ satisfying (1). It turns out that any searchable set (of total elements) is topologically compact, and, mimicking the Tychonoff theorem from topology, it was shown that searchable sets are closed under countable products. This relies on a *countable-product functional* of type

$$(JA)^n \to JA^n \quad (n \leq \omega),$$

which can be obtained by iterating the binary product of the monad J discussed above.

In [9], we considered much more general choices for A and R (objects of a cartesian closed category), and for ϕ (e.g. supremum functional when R are the reals in the category of sets, or in suitable categories of spaces). Moreover, we considered the above product in more generality, allowing the object A to vary, i.e. having type

$$\prod_{i<n} JA_i \to J\left(\prod_{i<n} A_i\right) \quad (n \leq \omega).$$

The case $n = \omega$ is restricted to a category of continuous maps of certain topological spaces, which include Kleene–Kreisel spaces of continuous functionals, and requires that the type R be topologically discrete to be well defined. This includes the natural numbers $\mathbb{N}$ and the Booleans $\mathbb{B} = \{0, 1\}$, of course, but also more general the types defined by induction in [6, Definition 4.12], for instance $R = ((\mathbb{N} \to \mathbb{B}) \to \mathbb{N})$. The need for discreteness is justified in [9, Remark 5.11].

We have shown that this iteration is an instance of the bar recursion scheme. In [7], we have established relations between this new form of bar recursion and the more traditional instances, such as Spector's bar recursion [16] and modified bar recursion [2,3].

In the present paper, we work with the category whose objects are formulas in HA^ω and morphisms are proofs of entailments, written in natural deduction style using λ-calculus notation [13], where we often regard the morphisms as realizers written in Gödel's system T. For the choices $T = J$ or $T = K$, or more generally any strong monad T, one has intuitionistic laws

$$\begin{array}{ll} T(A \to B) \to TA \to TB & \text{(functor)} \\ A \to TA & \text{(unit)} \\ TTA \to TA & \text{(multiplication)} \\ A \wedge TB \to T(A \wedge B) & \text{(strength).} \end{array}$$

In the terminology of [1], the construction T is a lax modal operator. It turns out that the infinite product of selection functions realizes, in the sense of modified realizability, the following shift principle for $T = J$, assuming that the type of realizers of the formula R is topologically discrete:

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

The well-known double negation shift is the case $T = K$ with $R = \bot$, but it is realized only for special types of formulas A, including those in the image of the negative translation, whereas the J-shift is realized for *all* formulas A. We also show that the double negation shift for formulas A in the image of a negative translation follows from the J-shift. With this, we will get an alternative way of interpreting classical analysis and extracting computational witnesses via infinite products of selection functions.

We plan to investigate the use of the product of selection functions for the extraction of computational content from proofs involving countable/dependent choice, as done by Seisenberger [14] with modified bar recursion. Based on the experimental results and theoretical conjectures of [6, Section 8.10] and [5], we conjecture that the product of selection functions will give rise to more efficient computational extraction of witnesses.

We stumbled upon the Peirce translation when studying products of selection functions in [9], after noting that the J construction is a monad and its algebras are formulas that satisfy Peirce law. The Peirce translation comes automatically out of this observation. Our aim here is to investigate the features of such a translation and the role of the product of selection functions on the interpretation of arithmetic and analysis.

Finally, let us briefly discuss the relation between the Peirce translation and the usual negative translations. First, the Peirce translation does not interpret ex-falso-quodlibet (efq), as most of the standard negative translations do. This can be viewed as a feature, as it gives a clear separation between classical reasoning (Peirce law) and the role of falsity (efq). In practice, however, this means that in order to apply the Peirce translation to classical proofs, we must first apply an "elimination of efq" procedure which takes us to minimal logic plus Peirce law (Theorem 5.2). It is then not hard to show that the Peirce translation when combined with the efq-elimination procedure is equivalent to the usual negative translation. It is in the interpretation of analysis that the Peirce translation comes out more naturally than the negative translation, as the J-shift can be interpreted for arbitrary formulas, whereas the K-shift only holds for a particular class of formulas (Theorem 4.3). And, for such formulas the K-shift follows from the J-shift (Proposition 4.2). In summary, we believe that the Peirce translation gives a conceptually cleaner explanation to the interpretation of the classical countable choice, but probably in practice, when applying the translation to concrete classical proofs, it might be better to use the standard negative translations.

This is a journal version of the paper [8]. We have improved the formulation and expanded several passages of the conference version, and included all proofs and the new Section 6.3 on weak König's lemma.

2. Preliminaries

2.1. Products of selection functions

As mentioned above, we use the infinite product of selection functions to interpret the classical countable and dependent choice. In this section, we briefly recall these product functionals which were first defined and studied in [9,7].

Definition 2.1 (*Products of Selection Functions*)**.** Given selection functions $\varepsilon \in JX$ and $\delta \in JY$, define their *product* $\varepsilon \otimes \delta \in J(X \times Y)$ as

$$(\varepsilon \otimes \delta)(p) = (a, b(a))$$

where

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

Similarly, given $\varepsilon \in JX$ and a family of selection functions $\delta \in X \to JY$, define their *dependent product* $\varepsilon \otimes_d \delta \in J(X \times Y)$ as

$$(\varepsilon \otimes_d \delta)(p) = (a, b(a))$$

where

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

We have also considered in [9] the functional obtained by iterating these binary products on an infinite sequence of selection functions.

Definition 2.2 (*Iterated Products of Selection Functions*)**.** The *iterated product of a family of selection functions* $\varepsilon \in \prod_{k \in \mathbb{N}} JX_k$ is defined in [9] by the equation

$$\mathsf{ps}_k(\varepsilon) \stackrel{J(\prod_{i \geq k} X_i)}{=} \varepsilon_k \otimes \left(\mathsf{ps}_{k+1}(\varepsilon)\right).$$

For $\varepsilon\colon \prod_{k \in \mathbb{N}}((\prod_{j<k} X_j) \to (JX_k))$ and $s\colon \Sigma_{k \in \mathbb{N}}(\prod_{j<k} X_j)$, define the *iterated dependent product of selection functions* as

$$\mathsf{PS}_s(\varepsilon) \stackrel{J(\prod_{i \geq k} X_i)}{=} \varepsilon_s \otimes_d \left(\lambda x^{X_k}.\mathsf{PS}_{s*x}(\varepsilon)\right).$$

The recursive definitions for ps and PS uniquely define functionals in the models of partial and total continuous functionals (cf. [9]). Finally, we remark that ps and PS are actually inter-definable over system T, as shown in [7].

2.2. Formal setting

Let ML stand for *minimal logic*, i.e. intuitionistic logic without the ex-falso-quodlibet axiom scheme $\mathsf{EFQ}\colon \perp \to A$ (see e.g. [18]). We denote by HA the formal system of Heyting arithmetic based on minimal logic, rather than intuitionistic logic.

$$
\dfrac{
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[A\to B]_\beta \quad [A]_\alpha}{B}}{R}\,(\mathrm{I})}{(A\to B)\to R}\,(\beta)}{A\to B}\,(\mathrm{III}) \quad [A]_\alpha}{B}}{R}\,(\mathrm{I})}{A\to R}\,(\alpha)}{A}\,(\mathrm{II})
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{[A\to B] \quad \dfrac{\dfrac{[A\to B]}{A\to R}\,(\mathrm{I})}{A}\,(\mathrm{II})}{B}}{R}\,(\mathrm{I})}{(A\to B)\to R}}{A\to B}\,(\mathrm{III})
}{B}
$$

Fig. 1. Derivation of Lemma 2.3(iii).

Given a formal system S, we write S^ω for the finite type generalization of S with a neutral treatment of equality (cf. [17]). Hence, Heyting arithmetic in all finite types is denoted by HA^ω. We use X, Y, Z for variables ranging over finite types.

Let us denote by *T-logic* the extension of ML with the *T-elimination axiom*

$$T\text{-}\mathsf{elim}\colon\ TA \to A.$$

Thus classical logic amounts to K-logic if we choose $R = \perp$ in the definition of K. Similarly, we refer to the extension of HA with the T-elimination axiom as *T-arithmetic* (TA). Then Peano arithmetic (PA) is K-arithmetic for $R = \perp$.

Although in HA^ω, one does not have dependent types, we will develop the rest of the paper working with types such as $\prod_{i\in\mathbb{N}} X_i$ rather than the special case X^ω, when all X_i are the same. The reason for this generalization is that the results developed below become clearer. Moreover, they go through for the more general setting where this simple form of dependent type is permitted. Nevertheless, we hesitate to define a formal extension of HA^ω with such dependent types, leaving this for future work. We believe that the techniques of Coquand and Spiwack [4] allow to generalize our results to Martin-Löf Type Theory, but we also leave this for future work.

We often write $\prod_i X_i$ for $\prod_{i\in\mathbb{N}} X_i$. If x has type X_n and s has type $\prod_{i<n} X_i$, then $s * x$ is the concatenation of s with x, which has type $\prod_{i\le n} X_i$. When $x\colon X_0$ and $\alpha\colon \prod_{i>0} X_i$ then $x * \alpha$ is the concatenation of x with the stream α, which has type $\prod_i X_i$. Moreover, $[\alpha](n)$ stands for the initial segment of the infinite sequence α of length n, i.e.

$$[\alpha](n) = \langle \alpha(0), \alpha(1), \ldots, \alpha(n-1)\rangle.$$

For a fixed formula R, we write $J_R A$ for $(A \to R) \to A$, i.e. the selection functions for A. Using this notation, the usual Peirce law corresponds to the principle of J-elimination

$$\mathsf{PL}_R\ :\ J_R A \to A.$$

We first observe that the construction $J_R A$ has the same properties as that of a strong monad (from category theory).

Lemma 2.3 (*Monad*). *The following are provable in* ML

(i) $A \to J_R A$
(ii) $J_R J_R A \to J_R A$
(iii) $J_R(A \to B) \to J_R A \to J_R B$.

Proof. All can be proved directly. Point (i) follows by weakening, while point (ii) makes use of three contractions over $A \to R$. The proof of (iii) is a bit trickier so we spell out the details here: assume that (I) $B \to R$ and (II) $J_R A$ and (III) $J_R(A \to B)$, we derive B as shown in Fig. 1. $\square$

2.3. Bar induction and continuity

Several proofs in the paper rely on two non-classical principles which we state here: the principle of *continuity*

$$\mathsf{CONT}\ :\ \forall q^{\prod_i X_i \to R}\forall\alpha\exists n\forall\beta([\alpha](n) \overset{\prod_{i\le n} X_i}{=} [\beta](n) \to q(\alpha) \overset{R}{=} q(\beta))$$

with R topologically discrete, and the scheme of *relativized quantifier-free bar induction* BI

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

where $Q(s)$ is an arbitrary predicate, $P(s)$ a quantifier free predicate in the language of HA^ω, and $\alpha \in Q$ and $s \in Q$ are shorthands for $\forall n Q([\alpha](n))$ and $Q(s)$ respectively.

3. *T*-translation

It is well known that several forms of the negative translation can be understood in terms of the continuation monad K. It is also well known that any monad T gives rise to a proof translation (see e.g. [1]). Here we consider the T-*translation* inductively defined as

$$
\begin{aligned}
P^T &= TP \\
(A \wedge B)^T &= A^T \wedge B^T \\
(A \vee B)^T &= T(A^T \vee B^T) \\
(A \to B)^T &= A^T \to B^T \\
(\exists x A)^T &= T(\exists x A^T) \\
(\forall x A)^T &= \forall x A^T.
\end{aligned}
$$

That is, we prefix T in front of atomic formulas, disjunctions and existential quantifications. For $T = K$ and $R = \perp$, this amounts to the standard Gödel–Gentzen negative translation [18], and for $R = A$, with A a Σ^0_1-formula, which corresponds to Friedman's A-translation [11] of the negative translation.

From well-known properties of monads on cartesian closed categories, one sees by induction that any C in the image of the T-translation is a T-algebra and in particular $TC \to C$ is provable. Putting all this together, we have that $TC \to C$ is provable in minimal logic, for formulas C in the image of the T-translation. For $T = K$ and $R = \perp$ the T-elimination principle $TC \to C$ amounts to double negation elimination. For $T = J$, this is the instance $((C \to R) \to C) \to C$ of Peirce law, and hence we also refer to the J-translation as the *Peirce translation*.

Because of the monad morphism $J \to K$, any K-algebra is a J-algebra, which gives the standard fact that the usual negative translations also eliminate Peirce law. Note that the implication $JA \to KA$ can be reversed if and only if $R \to A$. In fact, a main difference between the K-translation and the J-translation is that the former also eliminates ex-falso-quodlibet EFQ $(\perp \to A)$, whereas the latter is sound with respect to EFQ but does not eliminate it.

The following facts are well known (see e.g. [1]) and are easily proved by induction on formulas, although they are usually stated for intuitionistic logic rather than minimal logic.

Lemma 3.1. *For any strong monad T, assuming that $(TA)^T$ is equivalent to TA^T, we have*

1. $\mathsf{ML} \vdash TA^T \to A^T$.
2. $\mathsf{ML} + T\text{-}\mathsf{elim} \vdash A^T \to A$.
3. $\mathsf{ML} + T\text{-}\mathsf{elim} \vdash A$ *if and only if* $\mathsf{ML} \vdash A^T$.

The above lemma allows one to extract realizing functions for $\prod^0_2$-theorems in minimal arithmetic with Peirce law without ever going through intuitionistic logic. We will see later that the main obstacle for a realizability interpretation of classical logic is the EFQ, which says that a realizer for falsity must be turned into a realizer for an arbitrary formula. That forces all negated formulas to be empty of realizers and hence blocks any direct use of realizability to proofs in classical logic. The well-known remedy is to use Friedman's trick of the A-translation, which effectively eliminates EFQ and hence allows one to inject computational content into negated formulas. The next theorem shows that Friedman's trick is not necessary if one starts with a classical proof that does not make use of EFQ.

Theorem 3.2. *Assume that $P(x, y) \to R$ and that the variable y is not free in R. If*

$$\mathsf{ML} + J\text{-}\mathsf{elim} \vdash \forall x \exists y P(x, y)$$

then also

$$\mathsf{ML} \vdash \forall x \exists y P(x, y).$$

Proof. First, note that under the assumption $P(x, y) \to R$, we have

(i) $\mathsf{ML} \vdash JP(x, y) \to P(x, y)$,
(ii) $\mathsf{ML} \vdash J\exists y P(x, y) \to \exists y P(x, y)$.

If $\mathsf{ML} + J\text{-}\mathsf{elim} \vdash \forall x \exists y P(x, y)$ then $\mathsf{ML} + J\text{-}\mathsf{elim} \vdash \exists y P(x, y)$, and hence Lemma 3.1 gives $\mathsf{ML} \vdash J\exists y JP(x, y)$, which by (i) and (ii), implies that $\mathsf{ML} \vdash \exists y P(x, y)$. □

The first part of the next proposition shows that if multiple instances of J-elimination are used in a proof, for different parameters R, one can reduce to a single instance with the conjunction of all the parameters. For example, this can be applied to the above theorem if one needs to use several instances of Peirce law. The second part shows that the J- and K-translations coincide over intuitionistic logic.

Proposition 3.3. 1. $\mathsf{ML} + J_{R_0 \wedge R_1}\text{-}\mathsf{elim} \vdash J_{R_0}\text{-}\mathsf{elim} \wedge J_{R_1}\text{-}\mathsf{elim}$.

2. *For* $R \equiv \perp$, *we have that* $\mathsf{ML} + \mathsf{EFQ} \vdash A^K \leftrightarrow A^J$.

Proof. The first part is routine verification. The second part follows from Proposition 4.2. $\square$

Putting Theorem 3.2 and Proposition 3.3 together with obtain:

Corollary 3.4 (*Folklore*)**.** *Assume that* $P(x, y) \to R_i$*, for all* $0 \leq i \leq n$*, with* $y \notin \mathsf{FV}(R_i)$*. If*

$$\mathsf{ML} + \mathsf{PL}_{R_0} + \cdots + \mathsf{PL}_{R_n} \vdash \forall x \exists y P(x, y)$$

then also

$$\mathsf{ML} \vdash \forall x \exists y P(x, y).$$

Proof. Let $R :\equiv R_0 \wedge \cdots \wedge R_n$. First, note that $P(x, y) \to R_i$ implies $P(x, y) \to R$ and hence both (over ML)

(i) $J_R P(x, y) \to P(x, y)$

(ii) $J_R \exists y P(x, y) \to \exists y P(x, y)$.

Assuming $\mathsf{ML} + \mathsf{PL}_{R_0} + \cdots + \mathsf{PL}_{R_n} \vdash \forall x \exists y P(x, y)$ by Lemma 3.3 we get $\mathsf{ML} + \mathsf{PL}_R \vdash \exists y P(x, y)$. Theorem 3.1 then implies $\mathsf{ML} \vdash J_R \exists y J_R P(x, y)$, which by (i) and (ii) implies that $\exists y P(x, y)$ is provable in ML. $\square$

Remark 3.5 (*call/cc*)**.** The type of the continuation passing style translation of call/cc can be written as $JKX \longrightarrow KX$, an instance of Peirce law, as observed by Griffin [12]. Its λ-term can be reconstructed as follows:

1. KX is a K-algebra, with structure map $KKX \xrightarrow{\mu} KX$.
2. Because we have a morphism $J \longrightarrow K$, every K-algebra is a J-algebra:

$$JA \longrightarrow KA \xrightarrow{\alpha} A.$$

3. call/cc is what results for $A = KX$ and $\alpha = \mu$:

$$JKX \longrightarrow KKX \xrightarrow{\mu} KX.$$

3.1. Arithmetic

If a formula does not have occurrences of disjunction or existential quantification, its T-translation only prefixes T to atomic formulas, and hence the T-translations of the Peano axioms follow from the Peano axioms. Moreover, the T-translation of each instance of the induction axiom is again an instance of induction. This shows that the T-translation maps TA into HA.

4. Countable choice and shift principles

Contrary to arithmetic, discussed just above, the T-translation does not map $\mathsf{TA}^\omega + \mathsf{AC}_{\mathbb{N}}$ into $\mathsf{HA}^\omega + \mathsf{AC}_{\mathbb{N}}$, where $\mathsf{AC}_{\mathbb{N}}$ is the axiom of countable choice

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

and this failure applies to the particular cases $T = J$ and $T = K$ too. In fact, the T-translation of $\mathsf{AC}_{\mathbb{N}}$ is

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

which is not an instance of $\mathsf{AC}_{\mathbb{N}}$. In order to overcome this, the following was first observed by Spector [16] for the special case $T = K$ and $R = \perp$, where

$$T\text{-}\mathsf{shift}(A) \quad : \quad \forall n^{\mathbb{N}} TA(n) \to T \forall n A(n).$$

Proposition 4.1. $\mathsf{AC}_{\mathbb{N}} + T\text{-}\mathsf{shift} \vdash \mathsf{AC}_{\mathbb{N}}^T$.

Proof. Let us show that $\mathsf{HA}^\omega + \mathsf{AC}_{\mathbb{N}} + T\text{-}\mathsf{shift} \vdash \mathsf{AC}_{\mathbb{N}}^T$. Applying T-shift to the premise $\forall n T \exists x A^T(n, x)$ of $\mathsf{AC}_{\mathbb{N}}^T$, we deduce that $T \forall n \exists x A^T(n, x)$. Functoriality of T applied to $\mathsf{AC}_{\mathbb{N}}$ with A instantiated to A^T gives

$$T \forall n \exists x A^T(n, x) \to T \exists f \forall n A^T(n, fn),$$

and hence we get $T \exists f \forall n A^T(n, fn)$ by modus ponens, which is the conclusion of $\mathsf{AC}_{\mathbb{N}}^T$. $\square$

It follows from Lemma 3.1 and Proposition 4.1 that the T-translation maps the theory $\mathsf{TA}^\omega + \mathsf{AC}_{\mathbb{N}}$ into $\mathsf{HA}^\omega + \mathsf{AC}_{\mathbb{N}} + T\text{-}\mathsf{shift}$. In the context of the dialectica interpretation, Spector showed that a form of bar recursion, now known as *Spector bar recursion*, realizes the *double negation shift* (DNS), which amounts to the T-shift for $T = K$ and $R = \perp$. Moreover, via different forms of bar recursion with R a Σ_1^0 formula, it is shown in [2,3] how computational information can also be extracted via (modified) realizability from proofs in classical analysis in the presence of countable choice. But the K-shift is established only for formulas $\exists x A^K$ where A^K is in the image of the K-translation. Now, note that for any formula A^K, we have $\perp \to \exists x A^K$.

Proposition 4.2. *Over minimal logic, if $R \to A$ then J-shift$(A) \to K$-shift(A).*

Proof. We know that $JA \to KA$ for any A, and the assumption $R \to A$ is easily seen to give the converse, and hence $JA \leftrightarrow KA$. Note that if $KA \to JA$ holds, then $R \to A$, and hence the assumption $R \to A$ is optimal. □

Hence the following gives an alternative way of realizing the K-shift for the purpose of extracting witnesses from classical proofs with countable choice. The notions in the assumptions of the following theorem are defined in [3,17]. The restriction on R is needed for the infinite product to be well-defined [9], and note that it is fulfilled if R is Σ^0_1 or a Harrop formula.

Theorem 4.3 ($\mathsf{HA}^\omega + \mathsf{BI} + \mathsf{CONT}$). *If the type of realizers of the formula R is topologically discrete, then ps_0 mr J-shift(A).*

Proof. We fully prove a stronger result in Section 6.2. □

We emphasize that this theorem states that the infinite product functional *itself* realizes the shift principle, in the sense that the type of ps using dependent types, i.e. $\prod_i JA_i \to J\prod_i A_i$, directly corresponds to the logical formula J-shift(A). This is in contrast with the work discussed above, where the bar recursive functionals in question do not have the type of the principle they realize, and instead are *used in order to define* functionals that realize shift principles. For instance, modified bar recursion, when written with dependent types, has type

$$\prod_s ((A_{|s|} \to R) \to \prod_n A_n) \to K \prod_n A_n,$$

which does not correspond directly to the logical formula K-shift(A).

We regard as rather striking the fact that a functional that was originally introduced to mimic a theorem from topology in a computational setting, as discussed in the introduction, turns out to have a natural logical reading related to traditional work in proof theory, and we think that this deserves further investigation. In summary, the J-shift can be seen as a logical analogue of the Tychonoff theorem from topology.

Before moving to the treatment of dependent choice, let us observe that the following apparent generalization of the T-shift is equivalent over HA to the original formulation.

Proposition 4.4. *The T-shift principle is equivalent to the* course-of-values T-shift

$$T^c\text{-shift}(A) \quad : \quad \forall n(\forall k < n\, A(k) \to TA(n)) \to T\forall n A(n).$$

Proof. It is straightforward that T^c-shift implies the T-shift. Conversely, assume that $\forall n(\forall k < n\, A(k) \to TA(n))$. By the extension law $(B \to TC) \to (TB \to TC)$ of strong monads in a cartesian closed category and induction on n, we deduce that $\forall n(\forall k < n\, TA(k) \to TA(n))$. Hence $\forall n TA(n)$ by course-of-values induction, and the T-shift gives the desired result. □

The reason we formulate this course-of-values variant of T-shift is because T^c-shift is directly realizable by the iteration of the dependent product PS.

Theorem 4.5 ($\mathsf{HA}^\omega + \mathsf{BI} + \mathsf{CONT}$). *If the formula R has a discrete type of realizers then $\mathsf{PS}_{\langle\rangle}$ mr J^c-shift(A).*

In Section 6.2, we show that PS in fact also realizes a more general logical principle that implies full dependent choice. But first, let us discuss the simpler case of dependent choice for numbers.

5. Dependent choice for $\mathbb{N}$

We now compare TA^ω and HA^ω with respect to the axiom of *dependent choice*

$$\mathsf{DC}_X \quad : \quad \forall n^{\mathbb{N}}, x^X \exists y^X A_n(x, y) \to \forall x_0 \exists \alpha(\alpha_0 = x_0 \wedge \forall n A_n(\alpha_n, \alpha_{n+1})).$$

In this section, we focus on the simpler case when $X = \mathbb{N}$. In Section 6.2 below, we consider the general case.

Proposition 5.1. $\mathsf{DC}_{\mathbb{N}} + T$-shift $\vdash \mathsf{DC}^T_{\mathbb{N}}$.

Proof. The argument is essentially the same as that of Proposition 4.1, but one applies the T-shift twice, to move T outside two numerical universal quantifiers. □

Hence, the T-translation maps $\mathsf{TA} + \mathsf{DC}_{\mathbb{N}}$ into $\mathsf{HA} + \mathsf{DC}_{\mathbb{N}} + T$-shift. In general, however, when X is an arbitrary type, not just $\mathbb{N}$, the situation is subtler, because the T-shift will not be available for $T = J$ (let alone $T = K$). The case $T = K$ has been addressed in [2,3], and in Section 6.2 below, we address the case $T = J$ (which has the case $T = K$ as a corollary).

The following theorem (cf. Proposition 1 of [3]) shows how one can extract witnesses from proofs of $\prod^0_2$-statements in classical analysis via the J-translation and the J-shift (as opposed to via the negative translation and the double negation shift).

Theorem 5.2. *If*

$$\mathsf{PA}^\omega + \mathsf{AC}_{\mathbb{N}} + \mathsf{DC}_{\mathbb{N}} \vdash \forall x^X \exists n^{\mathbb{N}} P(x, n)$$

then one can extract a term t in system $\mathsf{T} + \mathsf{ps}$ *such that*

$$\mathsf{MA}^{\omega} + \mathsf{BI} + \mathsf{CONT} \vdash P(x, tx)$$

where MA^{ω} *denotes arithmetic in all finite types based on minimal logic.*

Proof. By prefixing each atomic formula with a double negation, EFQ is eliminated. Hence the assumption of the theorem implies

$$\mathsf{MA}^{\omega} + J_{\perp}\text{-elim} + \mathsf{AC}_{\mathbb{N}} + \mathsf{DC}_{\mathbb{N}} \vdash \forall x \exists n \neg\neg P(x, n).$$

Because the proof is in ML, we can replace $\perp$ by any formula, which we take to be $R \equiv \exists n P(x, n)$

$$\mathsf{MA}^{\omega} + J_R\text{-elim} + \mathsf{AC}_{\mathbb{N}} + \mathsf{DC}_{\mathbb{N}} \vdash \forall x \exists n((P(x, n) \to R) \to R).$$

Hence,

$$\mathsf{MA}^{\omega} + J_R\text{-elim} + \mathsf{AC}_{\mathbb{N}} + \mathsf{DC}_{\mathbb{N}} \vdash \forall x \exists n P(x, n).$$

By the J-translation, we have

$$\mathsf{MA}^{\omega} + \mathsf{AC}^J_{\mathbb{N}} + \mathsf{DC}^J_{\mathbb{N}} \vdash \forall x J \exists n J P(x, n),$$

and, by the choice of R, we have $J\exists n J P(x, n) \to P(x, n)$. Therefore,

$$\mathsf{MA}^{\omega} + \mathsf{AC}^J_{\mathbb{N}} + \mathsf{DC}^J_{\mathbb{N}} \vdash \forall x \exists n P(x, n).$$

Now we are done because $\mathsf{AC}^J_{\mathbb{N}}$ and $\mathsf{DC}^J_{\mathbb{N}}$ follow in $\mathsf{MA}^{\omega} + \mathsf{AC}_{\mathbb{N}} + \mathsf{DC}_{\mathbb{N}}$ from J-shift which, by Theorem 4.3, is realized by ps, and because $\mathsf{AC}_{\mathbb{N}}$ and $\mathsf{DC}_{\mathbb{N}}$ have simple modified realizability witnesses. □

6. Full dependent choice

We have discussed how one normally interprets the axiom of *countable* choice computationally by reducing it to the computational interpretation of the double negation shift (cf. [16,2,3] and Theorem 5.2 above). When it comes to the computational interpretation of the *dependent* choice

$$\mathsf{DC}\colon \forall n, x \exists y B_n(x, y) \to \forall x_0 \exists \alpha[\alpha 0 = x_0] \forall n B_n(\alpha n, \alpha(n+1)),$$

however, one normally does it directly, as it seems not possible to reduce the negative translation of DC using the simple double negation shift. In this section, continuing the discussion started in Section 4, we show that what is needed in order to approach this from a logical point of view is a *dependent* variant of the shift principle.

6.1. Weak dependent choice

We start our analysis, however, with the special case of the *weak dependent choice* wDC

$$\forall n^{\mathbb{N}} (\forall i < n\, \exists x^{X_i} A_i(x) \to \exists x^{X_n} A_n(x)) \to \exists \alpha \forall n A_n(\alpha(n)),$$

and the following generalization of the J-shift, which we call the *course-of-values* J-shift,

$$J^c\text{-shift}\colon \forall n(\forall i < n\, A(i) \to JA(n)) \to J\forall n A(n).$$

As shown in Proposition 4.4, the principle J^c-shift follows from J-shift by a simple application of course-of-values induction. The next lemma shows that the J-translation of the weak dependent choice wDC can be reduced to the standard countable choice plus J^c-shift.

Lemma 6.1. $\mathsf{AC}_{\mathbb{N}} + J^c\text{-shift} \vdash \mathsf{wDC}^J$.

Proof. Let $B(n)$ be $\exists x A_n(x)$. Assume that the premise of wDC^J holds, i.e.

$$\forall n(\forall i < n B(i) \to JB(n)).$$

By the course-of-values J-shift, we have $J\forall n B(n)$, that is, $J\forall n \exists x A_n(x)$. By $\mathsf{AC}_{\mathbb{N}}$, we obtain the conclusion of wDC^J. □

Theorem 6.2 $(\mathsf{HA}^{\omega} + \mathsf{BI} + \mathsf{CONT})$**.** $\mathsf{PS}_{\langle\rangle}$ mr J^c-shift.

We formulate and prove a stronger version of this in Theorem 6.4.

6.2. Full dependent choice

We can generalize wDC so that the witness for point n might depend on all witnesses A_i for $k < n$. Suppose A_n is a predicate on finite sequence $\prod_{k\le n} X_k$, then

$$\mathsf{DC}_{\mathsf{seq}} \ : \ \forall s(\forall j < |s|\, A_j([s](j+1)) \to \exists x A_{|s|}(s * x)) \to \exists \alpha \forall n A_n([\alpha](n+1)),$$

which we call the *dependent choice* for finite sequences. Essentially the same axiom was proposed by Monika Seisenberger in [14, Section 2.3]. For the sake of completeness, we include a proof that our formulation is equivalent to DC.

Lemma 6.3. $\mathsf{DC}_{\mathsf{seq}}$ *and* DC *are equivalent over* PA^ω.

Proof. First, let us show how $\mathsf{DC}_{\mathsf{seq}}$ can be used to prove the usual formulation of dependent choice. Consider

$$A_n^{x_0}(s) \equiv (|s| = n+1) \wedge (s_0 = x_0) \wedge \forall i < (|s|-1)\, B_i(s_i, s_{i+1}).$$

It is easy to show that the hypothesis $\forall n, x \exists y B_n(x, y)$ implies

$$\forall s(\forall i < |s|\, A_i^{x_0}([s](i+1)) \to \exists x A_{|s|}^{x_0}(s * x)).$$

Therefore, by $\mathsf{DC}_{\mathsf{seq}}$, we get $\exists \alpha \forall n A_n^{x_0}([\alpha](n+1))$, which implies

$$\exists \alpha(\alpha(0) = x_0 \wedge \forall n B_n(\alpha(n), \alpha(n+1))).$$

For the other direction, assume that a predicate $A_n(s)$ is given, such that the premise of $\mathsf{DC}_{\mathsf{seq}}$ holds. Define

$$B_n(s, t) \equiv (|s| = n) \to (|t| = n+1 \wedge [s](n) = [t](n) \wedge (\forall i < |s|\, A_i([s](i+1)) \to A_n(t))).$$

This says that if all non-empty initial segments of s satisfy A_i then t also satisfies A_n. The assumed premise of $\mathsf{DC}_{\mathsf{seq}}$ implies $\forall n \forall s \exists t B_n(s, t)$, which by DC gives

$$\forall s_0 \exists \alpha(\alpha(0) = s_0 \wedge \forall n B_n(\alpha(n), \alpha(n+1))).$$

Considering $s_0 = \langle\,\rangle$, we conclude that

$$\exists \alpha \forall n B_n(\alpha(n), \alpha(n+1)).$$

By construction of B_n, if we take $\beta(i) = (\alpha(i+1))_i$, we get a witness for the conclusion of $\mathsf{DC}_{\mathsf{seq}}$, as required. □

We now argue that $\mathsf{DC}_{\mathsf{seq}}$ is the natural generalization of the course-of-values J^c-shift, discussed in Section 6.1. Consider the binary case of J^c-shift

$$JA(0) \wedge (A(0) \to JA(1)) \to J(A(0) \wedge A(1)).$$

First, suppose that each $A(n)$ is a predicate on finite sequences of length n, i.e. of the form $A(n) = \exists s^{\prod_{i<n} X_i} B_n(s)$. We then have

$$J\exists s B_0(s) \wedge (\exists s B_0(s) \to J\exists t B_1(t)) \to J(\exists s B_0(s) \wedge \exists t B_1(s)).$$

We are interested in the case when the finite sequence witnessing B_n is required to be an extension of a finite sequence witnessing B_i, for $i < n$,

$$J\exists s B_0(s) \wedge \forall s(B_0(s) \to J\exists x B_1(s * x)) \to J\exists t(B_0([t](0)) \wedge B_1([t](1))).$$

The generalization of this to infinitely many predicates is precisely $\mathsf{DC}_{\mathsf{seq}}$.

Based on this observation, we now show that PS, which in Theorem 6.2 is claimed to realize J^c-shift, also realizes the J-translation of $\mathsf{DC}_{\mathsf{seq}}$ directly.

Theorem 6.4 ($\mathsf{HA}^\omega + \mathsf{BI} + \mathsf{CONT}$)**.** *Let R be a Σ^0_1-formula. Then* $\mathsf{PS}_{\langle\rangle}\ \mathsf{mr}\ \mathsf{DC}^J_{\mathsf{seq}}$.

Proof. Assume that the realizer for $\exists y^{Y_n} A_n(s * y)$ has type $X_n(s) \equiv \Sigma_{y \in Y_n} Z_n(s * y)$. Moreover, assume that we are given functionals ε and q such that

$$\varepsilon \quad \mathsf{mr} \quad \forall s(\forall i < |s|\, A_i([s](i+1)) \to J\exists y A_{|s|}(s * y))$$

$$q \quad \mathsf{mr} \quad \exists \alpha \forall n A_n([\alpha](n+1)) \to R.$$

Then ε and q have types

$$\prod_s \left(\prod_{i<|s|} Z_i([s](i+1)) \to JX_{|s|}(s) \right) \quad \text{and} \quad \sum_{\alpha \in \prod_i Y_i} \prod_n Z_n([\alpha](n+1)) \to R,$$

respectively. We need to show that

$$\mathsf{PS}_{\langle\rangle}(\varepsilon)(q) \;\mathsf{mr}\; \exists \alpha \forall n A_n([\alpha](n+1)).$$

For a sequence of pairs $t\colon \prod_{i<n}(V_i \times W_i)$, we write $t^0\colon \prod_{i<n} V_i$ for the projection of the sequence on all first elements. In what follows, t is a sequence of pairs where the first elements of each pair t^0 determine the type of the second elements of each pair, and hence t has the type $\prod_{i<n} X_i([t^0](i)) \equiv \prod_{i<n} \sum_{y \in Y_i} Z_i([t^0](i) * y))$. We prove $\forall t P(t)$ by relativized bar induction (cf. [3]), where

$$P(t) \equiv \mathsf{PS}_t(\varepsilon)(q_t) \;\mathsf{mr}\; \exists \alpha \forall n\, A_{|t|+n}(t^0 * [\alpha](n+1)).$$

The bar induction will be relativized to the predicate

$$Q(t) \equiv \forall i < |t| \; (t_i \;\mathsf{mr}\; \exists y A_i([t^0](i) * y)).$$

The first hypothesis $Q(\langle\rangle)$ of the bar induction is vacuously true. We now prove the two remaining hypotheses (i) and (ii).

(i) $\forall \alpha^Q \exists k P([\alpha](k))$. Given α satisfying Q, let k be a point of continuity of q at α (here we are using the discreteness of the type of realizers of R, which follows from the fact that R is a Σ^0_1-formula). We must show $P([\alpha](k))$, i.e.

$$\mathsf{PS}_{[\alpha](k)}(\varepsilon)(q_{[\alpha](k)}) \;\mathsf{mr}\; \exists \beta \forall n\, A_{k+n}(([\alpha](k))^0 * [\beta](n+1)).$$

Let $\langle \gamma, \delta \rangle = \mathsf{PS}_{[\alpha](k)}(\varepsilon)(q_{[\alpha](k)})$. The above follows from, for all n,

$$(\dagger) \quad \delta(n) \;\mathsf{mr}\; A_{k+n}(([\alpha](k))^0 * [\gamma](n+1)),$$

which we establish by course-of-values induction as follows. Unfolding the definition of PS, $(\dagger)$ is equivalent to

$$(\varepsilon_{[\alpha](k)*r}(\lambda x. q_{[\alpha](k)*r*x}(\mathsf{PS}_{[\alpha](k)*r*x}(\varepsilon)(q_{[\alpha](k)*r*x}))))_1 \;\mathsf{mr}\; A_{k+n}(([\alpha](k))^0 * [\gamma](n+1)),$$

where $r = [\mathsf{PS}_{[\alpha](k)}(\varepsilon)(q_{[\alpha](k)})](n)$ and $x\colon X_{k+n}([\alpha](k)*r)$. By the fact that k is a point of continuity of q at α, this is equivalent to

$$(\varepsilon_{[\alpha](k)*r}(\lambda x. q_{[\alpha](k)*r*x}(\mathbf{0})))_1 \;\mathsf{mr}\; A_{k+n}(([\alpha](k))^0 * [\gamma](n+1)).$$

Hence, by the assumption on ε, it remains to show that $[\alpha](k) * r \in Q$ and that

$$\lambda x. q_{[\alpha](k)*r*x}(\mathbf{0}) \;\mathsf{mr}\; \exists y^{Y_{k+n}} A_{k+n}(([\alpha](k))^0 * [\gamma](n) * y) \to R.$$

The first follows by the hypothesis of the course-of-values induction. The second follows from the assumptions on q using $[\alpha](k) * r \in Q$.

(ii) $\forall s^Q (\forall t, x(Q(s * t * x) \to P(s * t * x)) \to P(s))$. Let $s \in Q$ be given, and assume that

$$(1) \; \forall t, x(Q(s * t * x) \to P(s * t * x)).$$

We must show $P(s)$, i.e.

$$\mathsf{PS}_s(\varepsilon)(q_s) \;\mathsf{mr}\; \exists \alpha \forall n A_{|s|+n}(s^0 * [\alpha](n+1)).$$

Again let $\langle \gamma, \delta \rangle = \mathsf{PS}_s(\varepsilon)(q_s)$. It is enough to show that

$$(\mathsf{PS}_s(\varepsilon)(q_s)(n))_1 \;\mathsf{mr}\; A_{|s|+n}(s^0 * [\gamma](n+1)),$$

which, by the definition of PS is

$$(\varepsilon_{s*r}(\lambda x. q_{s*r*x}(\mathsf{PS}_{s*r*x}(\varepsilon)(q_{s*r*x}))))_1 \;\mathsf{mr}\; A_{|s|+n}(s^0 * [\gamma](n+1)),$$

where $r = [\mathsf{PS}_s(\varepsilon)(q_s)](n)$. This can be reduced to prove

$$(2) \; \lambda x. q_{s*r*x}(\mathsf{PS}_{s*r*x}(\varepsilon)(q_{s*r*x}))) \;\mathsf{mr}\; \exists y A_{|s|+n}(s^0 * [\gamma](n) * y) \to R.$$

Now, assume x is such that $Q(s * r * x)$. Then, by (1), we have $P(s * r * x)$, i.e.

$$(3) \; \mathsf{PS}_{s*r*x}(\varepsilon)(q_{s*r*x}) \;\mathsf{mr}\; \exists \alpha \forall n A_{|s*r*x|+n}((s * r * x)^0 * [\alpha](n+1)).$$

By the assumption on q, we have that (3) implies (2), which concludes the proof that $\forall t P(t)$, and the desired result follows by considering $t = \langle\rangle$. $\square$

Corollary 6.5. *If*

$$\mathsf{PA}^\omega + \mathsf{AC}_{\mathbb{N}} + \mathsf{DC}_{\mathsf{seq}} \vdash \forall x^X \exists n^{\mathbb{N}} P(x, n)$$

then one can extract a term t in system $\mathsf{T} + \mathsf{ps}$ *such that*

$$\mathsf{HA}^\omega + \mathsf{BI} + \mathsf{CONT} \vdash P(x, tx).$$

Proof. $\mathsf{AC}_{\mathbb{N}}$ and $\mathsf{DC}_{\mathsf{seq}}$ are modified-realizable in system T. The result follows because ps is inter-definable with PS (cf. [7]), and hence J^d-shift is modified-realizable in $T + \mathsf{ps}$. $\square$

6.3. Weak König's Lemma

It has been shown by the Reverse Mathematics program [15] that weak König's lemma,

$$\mathsf{WKL} \ : \ \forall n \exists s^{\mathbb{B}^*}(|s| = n \wedge T(s)) \to \exists \alpha^{\mathbb{B}^\omega} \forall n T([\alpha](n)),$$

is one of the most fundamental theorems in mathematics. Here $T(s)$ is assumed to be a $\prod_1^0$ predicate, and to be prefix-closed, i.e. $T(s * t) \to T(s)$. It is folklore that WKL can be proved using choice, although we have not been able to find a reference with an explicit formulation and proof of this. In this section, we show that WKL follows rather directly from $\mathsf{DC}_{\mathsf{seq}}$ of type $\mathbb{B}$ for $\prod_1^0$ formulas, and hence it can be easily interpreted using the interpretation of $\mathsf{DC}_{\mathsf{seq}}$ given above. Moreover, we observe that WKL in turn easily implies $\mathsf{DC}_{\mathsf{seq}}$ for $\prod_1^0$ formulas, so that these two principles are equivalent.

Proposition 6.6.

1. $\prod_1$-$\mathsf{DC}_{\mathsf{seq}}^{\mathbb{B}}$ *implies* WKL, *over* PA^ω.
2. WKL *implies* $\prod_1^0$-$\mathsf{DC}_{\mathsf{seq}}^{\mathbb{B}}$, *over* HA^ω.

Proof. (1): Given a $\prod_1$ predicate $T(s)$ assumed to satisfy

$$(*) \quad \forall n \exists s(|s| = n \wedge T(s))$$

we define another $\prod_1$ predicate

$$A_n(s) = (|s| = n + 1) \wedge \forall k \exists t(|t| = k \wedge T(s * t)).$$

Now, by classical logic, $(*)$ implies

$$\forall s(\forall j < |s|\, A_j([s](j + 1)) \to A_{|s|}(s * 0) \vee A_{|s|}(s * 1)).$$

By $\mathsf{DC}_{\mathsf{seq}}$, we have an α satisfying $\forall n A_n([\alpha](n + 1))$, i.e.

$$\exists \alpha \forall n \forall k \exists t(|t| = k \wedge T([\alpha](n + 1) * t)).$$

Taking $k = 0$, we obtain the conclusion of WKL, as $T(\langle\,\rangle)$ holds by $(*)$.

(2): For the other direction, given a $\prod_1$ predicate $A_n(s)$, where $s \colon \mathbb{B}^*$, we define a $\prod_1$-tree as

$$T(s) = \forall i < |s|\, A_i([s](i)).$$

$T(s)$ is the prefix-closure of $A_n(s)$. Moreover, assuming $A_n(s)$ satisfies the premise of $\mathsf{DC}_{\mathsf{seq}}$

$$\forall s(\forall j < |s|\, A_j([s](j + 1)) \to \exists x A_{|s|}(s * x))$$

one can show by induction that $T(s)$ satisfies the premise of WKL, i.e. the condition $(*)$ of previous proof. By WKL, we have $\exists \alpha \forall k T([\alpha](k))$, which by definition of $T(s)$ is

$$\exists \alpha \forall k \forall i < k\, A_i([\alpha](i)).$$

This clearly implies the conclusion of $\mathsf{DC}_{\mathsf{seq}}$. □

Acknowledgements

We thank Thomas Powell for carefully proofreading and spotting an error in a previous version of this paper. The second author also gratefully acknowledges the support of the Royal Society (grant 516002.K501/RH/kk).

References

[1] P. Aczel, The Russell–Prawitz modality, Mathematical Structures in Computer Science 11 (4) (2001) 541–554. Modalities in type theory (Trento, 1999).
[2] S. Berardi, M. Bezem, T. Coquand, On the computational content of the axiom of choice, The Journal of Symbolic Logic 63 (2) (1998) 600–622.
[3] U. Berger, P. Oliva, Modified bar recursion and classical dependent choice, Lecture Notes in Logic 20 (2005) 89–107.
[4] T. Coquand, A. Spiwack, A proof of strong normalisation using domain theory, Logical Methods in Computer Science 3 (4) (2007) 4:12–16.
[5] M.H. Escardó, Infinite sets that admit fast exhaustive search, in: Proceedings of LICS, 2007, pp. 443–452.
[6] M.H. Escardó, Exhaustible sets in higher-type computation, Logical Methods in Computer Science 4 (3) (2008) paper 4.
[7] M.H. Escardó, P. Oliva, Computational interpretations of analysis via products of selection functions, in: F. Ferreira, B. Lowe, E. Mayordomo, L.M. Gomes (Eds.), Computability in Europe 2010, in: LNCS, Springer, 2010, pp. 141–150.
[8] M.H. Escardó, P. Oliva, The Peirce translation and the double negation shift, in: F. Ferreira, B. Löwe, E. Mayordomo, L.M. Gomes (Eds.), Programs, Proofs, Processes — CiE 2010, in: LNCS, vol. 6158, Springer, 2010, pp. 151–161.
[9] M.H. Escardó, P. Oliva, Selection functions, bar recursion, and backward induction, Mathematical Structures in Computer Science 20 (2) (2010) 127–168.
[10] M.H. Escardó, P. Oliva, Sequential games and optimal strategies, Royal Society Proceedings A 467 (2011) 1519–1545.
[11] H. Friedman, Classically and intuitionistically provably recursive functions, in: D. Scott, G. Müller (Eds.), Higher Set Theory, in: Lecture Notes in Mathematics, vol. 669, Springer, Berlin, 1978, pp. 21–28.
[12] T.G. Griffin, A formulae-as-types notion of control, in: 17th Annual ACM Symp. on Principles of Programming Languages, POPL'90, San Francisco, CA, USA, 1990, pp 17–19.

[13] J. Lambek, P.J. Scott, Introduction to higher order categorical logic, in: Cambridge Studies in Advanced Mathematics, vol. 7, Cambridge University Press, Cambridge, 1986.
[14] M. Seisenberger, Programs from proofs using classical dependent choice, Annals of Pure and Applied Logic 153 (1–3) (2008) 97–110.
[15] S.G. Simpson, Subsystems of Second Order Arithmetic, in: Perspectives in Mathematical Logic, Springer, Berlin, 1999.
[16] 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 (Ed.), in: Recursive Function Theory: Proc. Symposia in Pure Mathematics, vol. 5, American Mathematical Society, Providence, Rhode Island, 1962, pp. 1–27.
[17] A.S. Troelstra, Metamathematical Investigation of Intuitionistic Arithmetic and Analysis, in: Lecture Notes in Mathematics, vol. 344, Springer, Berlin, 1973.
[18] A.S. Troelstra, H. Schwichtenberg, Basic Proof Theory, 2nd ed., Cambridge University Press, Cambridge, 2000.