A Direct Version of Veldman's Proof of Open Induction on Cantor Space via Delimited Control Operators

Danko Ilik, Keiko Nakata

A Direct Version of Veldman's Proof of Open Induction on Cantor Space via Delimited Control Operators*†

Danko Ilik[1] **and Keiko Nakata**[2]

1 **Research Center for Computer Science and Information Technologies**
Macedonian Academy of Sciences and Arts
Skopje, Macedonia
danko.ilik@gmail.com

2 **Institute of Cybernetics**
Tallinn University of Technology
Tallinn, Estonia
keiko@cs.ioc.ee

Abstract

First, we reconstruct Wim Veldman's result that Open Induction on Cantor space can be derived from Double-negation Shift and Markov's Principle. In doing this, we notice that one has to use a countable choice axiom in the proof and that Markov's Principle is replaceable by slightly strengthening the Double-negation Shift schema. We show that this strengthened version of Double-negation Shift can nonetheless be derived in a constructive intermediate logic based on delimited control operators, extended with axioms for higher-type Heyting Arithmetic. We formalize the argument and thus obtain a proof term that directly derives Open Induction on Cantor space by the shift and reset delimited control operators of Danvy and Filinski.

1998 ACM Subject Classification F.4.1 Mathematical Logic, F.3.3 Studies of Program Constructs

Keywords and phrases open induction, axiom of choice, double negation shift, Markov's principle, delimited control operators

Digital Object Identifier 10.4230/LIPIcs.TYPES.2013.188

1 Introduction

Let X be a set with an equality relation $=_X$ and a binary relation $<_X$. We denote by X^ω and X^* the set of infinite sequences, or *streams*, over X and the set of finite sequences over X, respectively. Let elements of X^ω be denoted by Greek letters α, β, γ, let natural numbers be denoted by n, k, l, m, and let $\overline{\alpha}n$ denote the finite sequence $\langle \alpha(0), \alpha(1), \ldots, \alpha(n-1) \rangle$, i.e., the initial segment of length n of the sequence α.

The lexicographic extension $<_{X^\omega}$ of $<_X$ is a binary relation on streams, defined by

$$\alpha <_{X^\omega} \beta \text{ iff } \exists n(\overline{\alpha}n =_{X^*} \overline{\beta}n \wedge \alpha(n) <_X \beta(n)),$$

where $=_{X^*}$ denotes the equality relation induced from $=_X$ by element-wise comparison, i.e., $p =_{X^*} q$ iff p and q are of the same length and element-wise equal with respect to $=_X$.

* D. Ilik's work is covered by a Kurt Gödel Research Prize Fellowship 2011.
† K. Nakata acknowledges the ERDF funded EXCS project, the Estonian Ministry of Education and Research research theme no. 0140007s12, and the Estonian Science Foundation grant no. 9398.

A non-empty subset U of X^ω is called *open* if there is an enumeration $\pi : \mathbb{N} \to X^*$ which can approximate U, in the sense that membership in U can be defined[1] by

$\alpha \in U$ iff $\exists n \exists k(\overline{\alpha}n =_{X^*} \pi(k))$.

The *Principle of Open Induction on* X^ω (equipped with $<_X$ and $=_X$) is the following statement, for U open:

$$\forall \alpha\, (\forall \beta <_{X^\omega} \alpha\, (\beta \in U) \to \alpha \in U) \to \forall \alpha (\alpha \in U). \tag{OI-X}$$

One immediately sees that OI-X has the form of a well-founded induction principle. However, one should note that, even for the simple choice of $X = \{0, 1\}$ equipped with the usual decidable order and equality relation, an open set U is generally uncountable, and the lexicographic ordering $<_{X^\omega}$ is not well-founded!

The utility of this principle has been recognized by Raoult [15] who gave, using OI-X, a new version of Nash-Williams' proof of Kruskal's theorem that does not explicitly use the Axiom of Dependent Choice[2].

OI-X was introduced in the context of Constructive Mathematics by Coquand [4]. He proved OI-X by relativized Bar Induction, and also first considered separately the version for X^ω being the Cantor space [5].

Berger [3] showed that OI-X in higher-type Arithmetic, where X can be any type ρ, is classically equivalent to the Axiom of Dependent Choice (DC) for the type ρ. He also gave a modified realizability interpretation of OI-X by a schema of Open Recursion, and showed that, unlike DC, OI-X is closed under double-negation- and A-translation – this means that there is a simple way to extract open-recursive programs from classical proofs of Π^0_2-statements that use DC or OI-X.

In the context of Constructive Reverse Mathematics, in a series of lectures [18], Veldman showed that Open Induction for Cantor space is equivalent to Double-negation Shift,

$$\forall n \neg\neg A(n) \to \neg\neg \forall n A(n) \quad \text{(for any formula } A(n)\text{)}, \tag{DNS}$$

in presence of Markov's Principle,

$$\neg\neg \exists n A_0(n) \to \exists n A_0(n) \quad \text{(for a decidable } A_0(n)\text{)}. \tag{MP}$$

Given that it is possible to obtain proofs for both MP [9] and DNS [11] using constructive logical systems based on delimited control operators, it is a natural next step to attempt to provide a direct constructive proof of OI for Cantor space based on delimited control operators. This is what we do in this paper.

The remainder of the paper is organized as follows. In Section 2, we reconstruct in detail Veldman's argument that proves OI on Cantor space from DNS and MP via the principle EnDec. In Section 3, we recall the logical system $\mathrm{MQC}_+(S)$ from [11] that is able to prove a strengthened version DNS_S of DNS using delimited control operators. DNS_S allows us to prove (a minimal logic version of) EnDec without explicitly using MP. In Section 4, we give a formalized proof term for OI on Cantor space in a variant of HA^ω based on the logical system $\mathrm{MQC}_+(S)$. In the concluding Section 5, we explain the current limitation of our approach for extracting proofs from programs and we mention directly related works.

[1] For simplicity, we exclude the possibility of $U = \emptyset$, so that we may take *total* enumerations π, rather than partial enumerations, sending $\mathbb{N}$ to $\mathrm{option}(X^*)$.

[2] Raoult proves OI-X using Zorn's Lemma.

2 From DNS and MP to Open Induction for Cantor Space

We will consider the case $X = \mathbb{B}$, where $\mathbb{B} = \{0, 1\}$ with $0 <_{\mathbb{B}} 1$ and $0 =_{\mathbb{B}} 0$, $1 =_{\mathbb{B}} 1$, that is, Open Induction *on Cantor space*, OI-$\mathbb{B}$. We will show that OI-$\mathbb{B}$ is provable from DNS, MP, and AC!$^{0,\mathbb{B}}$, where

$$\forall x^{\mathbb{N}} \exists! y^{\mathbb{B}} A(x, y) \to \exists f^{\mathbb{N}\to\mathbb{B}} \forall x^{\mathbb{N}} A(x, f(x)) \tag{AC!$^{0,\mathbb{B}}$}$$

is a restriction of the Axiom of Unique Countable Choice (also known as Countable Comprehension). All the arguments of this section take place in plain intuitionistic logic; if a principle that is not intuitionistically derivable is used, that is explicitly noted.

In addition to the already introduced notational conventions, let p, q, r, s denote finite binary sequences (bit-strings), $\mathbb{B}^*$, and let $p * q$ denote the concatenation of p and q. For a natural number k, $\mathbb{B}^k$ denotes the set of bit-strings of length k. Concrete bit-strings are constructed using the notation $\langle\cdot\rangle$, e.g. $\langle\rangle$ denotes an empty sequence, $\langle 0\rangle$ the bit-string of length 1 that contains a 0, $\langle 1, 1, 1, 1\rangle$ the bit-string that contains four 1's, etc. Thus $p * \langle 0\rangle$ means that a zero bit is appended at the end of p. The function $\mathrm{len}(p)$ computes the length of p. Analogously to the initial segment function $\overline{\alpha}n$ on infinite sequences, we denote by $\overline{p}n$ the initial segment function on finite sequences, with default value $\overline{p}n := p$ when $n > \mathrm{len}(p)$. Instead of writing $<_{\mathbb{B}^\omega}$ and $=_{\mathbb{B}^*}$, we simply write $<$ and $=$. We abbreviate $(S_1 \to S_2) \wedge (S_2 \to S_1)$ to $(S_1 \leftrightarrow S_2)$. We may write $n \notin A$ to mean $\neg(n \in A)$.

By a Σ-*formula*, we mean a formula built only from existential quantifiers (over the set $\mathbb{N}$), disjunction, conjunction, and the equality symbol "=" for $\mathbb{N}$. This definition is equivalent to the usual definition of Σ^0_1-formula if the language has all the primitive recursive symbols, as is the case for the system from Section 4.

We say that a set $B \subseteq \mathbb{N}$ is *enumerable* when the membership in B is a Σ-formula, i.e., $n \in B$ is defined as $S(n)$ for a Σ-formula S. Equivalently[3], B is enumerable when B is given by a function $f : \mathbb{N} \to \mathbb{N}$ such that $n \in B$ is a notation for $\exists m(f(m) = n + 1)$. A set $B \subseteq \mathbb{N}$ is *decidable* when we have that $\forall n(n \in B \vee n \notin B)$[4].

Veldman introduced the following principle.

▶ **Axiom 1** (EnDec)**.** Assume $B \subseteq \mathbb{N}$ is enumerable. Let, for any decidable $C \subseteq B$, we have that, if $\exists m(m \notin C)$, then $\exists m(m \notin C \wedge m \in B)$. Then $\mathbb{N} \subseteq B$ (and hence B is decidable).

Note that EnDec holds classically, since classically any B is decidable, so we may set $C := B$ to obtain $\mathbb{N} \subseteq B$. Our interest in EnDec here is because it is a stepping stone to proving OI-$\mathbb{B}$.

▶ **Theorem 1.** *Assuming AC!$^{0,\mathbb{B}}$, EnDec implies Open Induction on Cantor space.*

Proof. Let A be a non-empty open subset of Cantor space[5] i.e., there exists $\pi : \mathbb{N} \to \mathbb{B}^*$ such that "$\alpha \in A$" is a notation for $\exists l, m(\overline{\alpha}l = \pi(m))$. Let also A be *progressive*, that is,

$$\forall \alpha(\forall \beta < \alpha(\beta \in A) \to \alpha \in A).$$

We want to show that $\forall \alpha(\alpha \in A)$. Define $B \subseteq \mathbb{B}^*$ as

$$p \in B \text{ iff } \exists k \forall q \in \mathbb{B}^k \exists l, m(\overline{p * q}\, l = \pi(m))$$

[3] "Equivalent" in the system from Section 4.
[4] In some literature, our "decidable" is called "detachable".
[5] The progressiveness on Cantor space in fact ensures that A is non-empty.

such that p is in B if p is "uniformly barred" by π. That is, $p \in B$ if there exists k such that any extension of p by a finite bit-string of length k is covered by $\pi(m)$ for some m[6].

It suffices to show $\langle\rangle \in B$ for the empty bit-string $\langle\rangle$, since we then know that π covers the entire Cantor space. We show that B is actually equal to $\mathbb{B}^*$, using EnDec. Notice that $\mathbb{B}^*$ is bijective to $\mathbb{N}$ by primitive recursive functions and B is enumerable[7], hence we may transport EnDec from $\mathbb{N}$ to $\mathbb{B}^*$. It is left to show that, for any decidable subset $C \subseteq B$, if $\exists q(q \notin C)$, then $\exists r(r \notin C \wedge r \in B)$.

Suppose that such C and q are given. If $\langle\rangle \in C \subseteq B$, then we have that $q \in B$. So we are done. We assume $\langle\rangle \notin C$. Since C is decidable, we can construct α, using AC!$^{0,\mathbb{B}}$, such that

$$\alpha(n) := \begin{cases} 0 & \text{, if } \overline{\alpha}n * \langle 0\rangle \notin C \\ 1 & \text{, if } \overline{\alpha}n * \langle 0\rangle \in C \text{ and } \overline{\alpha}n * \langle 1\rangle \notin C \\ 0 & \text{, if } \overline{\alpha}n * \langle 0\rangle \in C \text{ and } \overline{\alpha}n * \langle 1\rangle \in C \end{cases}$$

The sequence α tries to stay outside of C for as long as possible and tries to be minimal. It first tries to "turn left" (value 0). If it was not possible, i.e., $\overline{\alpha}n * \langle 0\rangle \in C$, then it tries to "turn right" (value 1). If neither was possible, then it defaults to "turning left". One may notice that if α fails to stay outside of C at $n+1$, i.e., $\overline{\alpha}n * \langle 0\rangle \in C$ and $\overline{\alpha}n * \langle 1\rangle \in C$, then we have $\overline{\alpha}n \in B$. This fact, a manifestation of the compactness of Cantor space, will be used later in the proof.

Now, we can find a prefix of α that is in B but not in C, by following α up to the first point where it enters B. Let us first prove that α is in A, which guarantees that α has a prefix in B, hence that α will enter B. We use progressiveness of A. Let $\beta < \alpha$ i.e., $\exists n(\overline{\beta}n = \overline{\alpha}n \wedge \beta(n) = 0 \wedge \alpha(n) = 1)$. We have to show $\beta \in A$. By construction of α, $\alpha(n) = 1$ is only possible if $\overline{\alpha}n * \langle 0\rangle \in C$ and $\overline{\alpha}n * \langle 1\rangle \notin C$. Noticing that $\overline{\beta}(n+1) = \overline{\beta}n * \langle 0\rangle = \overline{\alpha}n * \langle 0\rangle$, this yields $\overline{\beta}(n+1) \in C \subseteq B$. We conclude that $\beta \in A$, which was to be shown.

From $\alpha \in A$, we obtain l, m such that $\overline{\alpha}l = \pi(m)$. We finish the proof by proving the following more general statement by induction

$$\forall n \leq l\,(\overline{\alpha}(l-n) \notin C \to \exists l'(\overline{\alpha}l' \notin C \wedge \overline{\alpha}l' \in B))\,.$$

Indeed, since we have $\langle\rangle \notin C$, by instantiating the above statement with $n := l$, we obtain p such that $p \notin C$ and $p \in B$.

In the base case, $n = 0$, we have that $\overline{\alpha}l \notin C$ by the hypothesis and that $\overline{\alpha}l \in B$ (from $\alpha \in A$); so we set $l' := l$. In the induction case for $n+1$ we consider three possibilities:

1. if $\overline{\alpha}(l-(n+1)) * \langle 0\rangle \notin C$, then $\overline{\alpha}(l-n) = \overline{\alpha}(l-(n+1)+1) = \overline{\alpha}(l-(n+1)) * \langle 0\rangle \notin C$ and we close the case by induction hypothesis;
2. similarly, if $\overline{\alpha}(l-(n+1)) * \langle 0\rangle \in C$ and $\overline{\alpha}(l-(n+1)) * \langle 1\rangle \notin C$, then $\overline{\alpha}(l-n) = \overline{\alpha}(l-(n+1)+1) = \overline{\alpha}(l-(n+1)) * \langle 1\rangle \notin C$, and we close the case by induction hypothesis;
3. if $\overline{\alpha}(l-(n+1)) * \langle 0\rangle \in C$ and $\overline{\alpha}(l-(n+1)) * \langle 1\rangle \in C$, then we get that $\overline{\alpha}(l-(n+1)) \in B$ as we noted earlier. Recalling that we also have $\overline{\alpha}(l-(n+1)) \notin C$ by hypothesis, we can set $l' := l-(n+1)$.

The first two cases could be merged into one, verifying only whether $\overline{\alpha}(l-(n+1)+1) \notin C$. ◀

---

[6] A bit-string p is *covered* by q if, as a bit-string, q is a prefix of p, or the open set given by p is covered by the open set given by q.

[7] B is enumerable because it is defined by a Σ-formula: the bounded universal quantifier "$\forall q \in \mathbb{B}^k$" does not pose a problem, since it could be interpreted as a bounded minimization operator, for example like in §3.5 of [12].

▶ Remark. In the previous proof, we used $\mathrm{AC!}^{0,\mathbb{B}}$ when constructing the sequence α by course-of-values recursion using the choice function extracted from the decidability of C. Since the principle EnDec is classically valid, not using a choice axiom would mean that one can reduce OI-$\mathbb{B}$ (and, using Berger's results [3], also Dependent Choice for $\mathbb{B}$) to plain classical logic without choice[8].

We now consider the principle of Double-negation Shift (DNS), which is independently important because it allows to interpret the double-negation translation of the Axiom of Countable Choice [16]. Following Veldman, we find it useful to consider the following variant of DNS.

▶ **Axiom 2** ($\mathsf{DNS}^{\mathsf{V}}$). $\neg\neg\forall n(A(n) \vee \neg A(n))$, for any formula $A(n)$.

▶ Remark. The proof of equivalence between DNS and $\mathrm{DNS}^{\mathrm{V}}$ is analogous to the proof of equivalence between the law of double-negation elimination (DNE) and the law of excluded middle (EM). In minimal logic, which is intuitionistic logic without the rule of $\bot$-elimination (*ex falso quodlibet*), EM is weaker than DNE [1]. We expect a similar result for DNS, i.e., that $\mathrm{DNS}^{\mathrm{V}}$ is weaker than DNS in minimal logic.

When quantifier-free formulas and decidable formulas coincide, as in Arithmetic, we may state Markov's Principle using Σ-formulas.

▶ **Axiom 3** (MP). For any Σ-formula S, we have that $\neg\neg S \to S$.

We can now prove EnDec from $\mathrm{DNS}^{\mathrm{V}}$and MP.

▶ **Theorem 2.** *$DNS^V$and MP together imply EnDec.*

Proof. Let the premises of EnDec hold. Given $n \in \mathbb{N}$, we have to prove $n \in B$, which is a Σ-formula. We are entitled to apply MP. Now, we have to show that $\neg\neg(n \in B)$. Suppose $\neg(n \in B)$. Thanks to $\mathrm{DNS}^{\mathrm{V}}$, it suffices to prove $\bot$ assuming moreover that B is decidable, i.e., $\forall n(n \in B \vee \neg(n \in B))$. We use the premise of EnDec by taking $C := B$ and recalling that we have $\neg(n \in B)$. This gives us $\exists m(m \in B \wedge \neg(m \in B))$, from which we derive $\bot$. ◀

3 A Constructive Logic Proving EnDec

In this section, we recall the logical system $\mathrm{MQC}_+(S)$ from [11], and show that EnDec is provable in $\mathrm{MQC}_+(S)$ (with a suitably instantiated parameter S), without an explicit use of MP, thanks to the slightly stronger form of DNS that $\mathrm{MQC}_+(S)$ proves.

$\mathrm{MQC}_+(S)$ is a pure predicate logic system, parameterized over a closed Σ-formula S, that, in addition to the usual rules of minimal intuitionistic predicate logic, adds two rules for proving the Σ-formula S [9]. The rule "reset",

$$\frac{\Gamma \vdash_S S}{\Gamma \vdash_\diamond S}\ \#\ (\text{"reset"}),$$

sets a marker (under the turnstile) meaning that one wants to prove S. Once the marker is set, one can use the "shift" rule,

[8] Classically $\mathrm{AC!}^{0,\mathbb{B}}$ is equivalent to Dependent Choice for $\mathbb{B}$ (in Berger's formulation), hence that we only use $\mathrm{AC!}^{0,\mathbb{B}}$ is not a concern.

[9] In the context of $\mathrm{MQC}_+(S)$, Σ-formulas coincide with formulas without $\forall$ and $\to$.

$$\frac{\Gamma, A \Rightarrow S \vdash_S S}{\Gamma \vdash_S A}\,\mathcal{S}\ (\text{“shift”}),$$

to prove by a principle related to double-negation elimination from classical logic. The idea is to internalize in the formal system the fact, known from Friedman-Dragalin's A-translation, that a classical proof of a Σ^0_1-formula can be translated to an intuitionistic proof of the same formula, showing that classical proofs of such formulas are in fact constructive. The first system built around this internalization idea was Herbelin's [9] with the power to derive Markov's Principle. It satisfies, like $\mathrm{MQC}_+(S)$, the disjunction and existence properties, characteristic of plain intuitionistic logic.

The names “shift” and “reset” come from the computational intention behind the normalization of these proof rules, Danvy and Filinski's delimited control operators [6, 7, 8]. These operators were developed in the theory of programming languages with the aim of enabling to write continuation-passing style (CPS) programs in so-called *direct style*. Since CPS transformations are known to be one and the same thing as double-negation translations [14], one can think of shift/reset in Logic as enabling to prove *directly* theorems whose double-negation translation is intuitionistically provable. In order for this facility to remain constructive, we allow its use only for proving Σ-formulas.

The natural deduction system for $\mathrm{MQC}_+(S)$ is given in Table 1 with proof term annotations. The diamond in the subscript of $\vdash$ is a wild-card: $\vdash_\diamond$ denotes either $\vdash$ or $\vdash_S$, where in the latter the subscript S is the same formula as the parameter S. We mark $\vdash$ with the parameter to record that a reset has been set. The rules should be read bottom-up, so that the marker is propagated from below to above the line. The usual intuitionistic rules neither “read” nor “write” this marker, hence $\diamond$ denotes the same below and above the line. The reset rule is the one that sets the marker (if it is not already set). If the marker has been already set, then the marker is simply kept. This kind of use of reset would have no logical purpose, but it would affect the course of normalization, hence the computational behavior of the proof term. The rule shift can only be applied when the marker is set, hence it is assured that we are ultimately proving the Σ-formula S.

The following theorem shows a utility of proving with shift and reset.

▶ **Theorem 3.** *Let S be a closed Σ-formula and $A(x)$ an arbitrary formula. The following version of DNS^V,*

$$\Big(\big(\forall x\big(A(x) \vee (A(x) \to S)\big)\big) \to S\Big) \to S, \tag{$\mathrm{DNS}^{\mathrm{V}}_S$}$$

is provable in $\mathrm{MQC}_+(S)$.

Proof. Using the proof term $\lambda h.\#h\Big(\tilde{\lambda}x.\mathcal{S}k.k\Big(\iota_2\big(\lambda a.k(\iota_1 a)\big)\Big)\Big)$. ◀

$\mathrm{DNS}^{\mathrm{V}}_S$ is a version of $\mathrm{DNS}^{\mathrm{V}}$, in which $\bot$ is generalized to a closed Σ-formula S. $\mathrm{DNS}^{\mathrm{V}}_S$ already has some form of MP built in, as can be seen from the proof of Theorem 4 below.

We now state a version of EnDec which is suitable for use in minimal logic, where $\bot$-elimination is absent.

▶ **Axiom 4** (A minimal-logic version of Axiom 1)**.** Assume that $B \subseteq \mathbb{N}$ is enumerable and $n \in \mathbb{N}$. Let, for any $s \in \mathbb{N}$ and any $C \subseteq B$, such that

$$\forall x\,(x \in C \vee (x \in C \to s \in B))\,,$$

Table 1 Natural deduction system for $\mathrm{MQC}_+(S)$, parameterized over a closed Σ-formula S, with proof terms annotating the rules.

$$\frac{(a : A) \in \Gamma}{\Gamma \vdash_\diamond a : A}\ \mathrm{Ax}$$

$$\frac{\Gamma \vdash_\diamond p : A_1 \qquad \Gamma \vdash_\diamond q : A_2}{\Gamma \vdash_\diamond (p, q) : A_1 \wedge A_2}\ \wedge_I \qquad \frac{\Gamma \vdash_\diamond p : A_1 \wedge A_2}{\Gamma \vdash_\diamond \pi_i\, p : A_i}\ \wedge^i_E$$

$$\frac{\Gamma \vdash_\diamond p : A_i}{\Gamma \vdash_\diamond \iota_i\, p : A_1 \vee A_2}\ \vee^i_I$$

$$\frac{\Gamma \vdash_\diamond p : A_1 \vee A_2 \qquad \Gamma, a_1 : A_1 \vdash_\diamond q_1 : C \qquad \Gamma, a_2 : A_2 \vdash_\diamond q_2 : C}{\Gamma \vdash_\diamond \mathsf{case}\ p\ \mathsf{of}\ \ (a_1.q_1 \| a_2.q_2) : C}\ \vee_E$$

$$\frac{\Gamma, a : A_1 \vdash_\diamond p : A_2}{\Gamma \vdash_\diamond \lambda a.p : A_1 \to A_2}\ \to_I \qquad \frac{\Gamma \vdash_\diamond p : A_1 \to A_2 \qquad \Gamma \vdash_\diamond q : A_1}{\Gamma \vdash_\diamond p\, q : A_2}\ \to_E$$

$$\frac{\Gamma \vdash_\diamond p : A(x) \qquad x\ \text{fresh}}{\Gamma \vdash_\diamond \tilde{\lambda} x.p : \forall x A(x)}\ \forall_I \qquad \frac{\Gamma \vdash_\diamond p : \forall x A(x)}{\Gamma \vdash_\diamond p\, t : A(t)}\ \forall_E$$

$$\frac{\Gamma \vdash_\diamond p : A(t)}{\Gamma \vdash_\diamond (t, p) : \exists x.A(x)}\ \exists_I$$

$$\frac{\Gamma \vdash_\diamond p : \exists x.A(x) \qquad \Gamma, a : A(x) \vdash_\diamond q : C \qquad x\ \text{fresh}}{\Gamma \vdash_\diamond \mathsf{dest}\ p\ \mathsf{as}\ (x.a)\ \mathsf{in}\ q : C}\ \exists_E$$

$$\frac{\Gamma \vdash_S p : S}{\Gamma \vdash_\diamond \#p : S}\ \#\ (\text{“reset”}) \qquad \frac{\Gamma, k : A \to S \vdash_S p : S}{\Gamma \vdash_S \mathcal{S}k.p : A}\ \mathcal{S}\ (\text{“shift”})$$

we have that, if

$$\exists m(m \in C \to s \in B),$$

then

$$\exists m((m \in C \to s \in B) \wedge m \in B).$$

Then, $n \in B$.

The following result is the minimal-logic analogue of Theorem 2, showing that an instance of Axiom 4 is derivable in $\mathrm{MQC}_+(S)$.

▶ **Theorem 4.** *Assume that $B \subseteq \mathbb{N}$ is enumerable and $n \in \mathbb{N}$. The instance of Axiom 4 with conclusion $n \in B$ is derivable in the system $\mathrm{MQC}_+(n \in B)$.*

Proof. Let the premises of Axiom 4 hold. To show that $n \in B$, which is a Σ-formula, we use $\mathrm{DNS}^{\mathrm{V}}_S$ for $A(x) := x \in B$ and $S := n \in B$. Now, given $\forall x(x \in B \vee (x \in B \to n \in B))$, we have to show $n \in B$. We use the premise of Axiom 4 for $s := n$ and $C := B$, and, using the trivial proof of $\exists m(m \in B \to n \in B)$ for $m := n$, the premise gives us a proof of $\exists m(m \in B \wedge (m \in B \to n \in B))$, from which we derive $n \in B$. ◀

4 A Proof Term for Open Induction

In this section, we give a proof term for OI on Cantor space in the system $\mathrm{HA}^{\omega}_{+}(S)$ (by suitably instantiating the parameter S), which is the system of axioms HA^{ω} (from §§1.6.15 of [17]) and $\mathrm{AC!}^{0,\mathbb{B}}$ added on top of the predicate logic $\mathrm{MQC}_{+}(S)$ — the need of $\mathrm{AC!}^{0,\mathbb{B}}$ is justified by Remark 2. Basic ingredients to construct the proof term are at hand: Theorem 1 and Theorem 4. We are to interpret them in $\mathrm{HA}^{\omega}_{+}(S)$ and combine the thus obtained proof terms for Theorem 1 and Theorem 4.

4.1 The system $\mathrm{HA}^{\omega}_{+}(S)$

Let S be a closed Σ-formula. First, we take a multi-sorted version of $\mathrm{MQC}_{+}(S)$, that is, given different sorts (denoted by $\sigma, \rho, \tau, \delta$), the language is extended with individual variables (denoted by x, y, z) of any sort, and quantifiers for all sorts. We will not annotate quantifiers with their sorts, since those will be clear from the context; we may annotate variables by their sorts when we want to avoid ambiguity.

The sorts are built inductively, according to the following rules: there is a sort named 0; if ρ and σ are sorts, then there is a sort named $\rho \to \sigma$. The intended interpretation is that the sort 0 stands for $\mathbb{N}$, the sort $0 \to 0$ stands for functions $\mathbb{N} \to \mathbb{N}$, the sort $((0 \to 0) \to 0)$ for functionals $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$, etc. We will employ the word 'type' instead of sort, henceforth, and we abbreviate the type $0 \to 0$ by 1.

Now, we add to the language a binary predicate symbol $=$ for individual terms of type 0, intended to be interpreted as (the decidable) equality on $\mathbb{N}$. We emphasize that we only have decidable equality. The individual terms will be built from the function symbols 0^0 (zero), $(\cdot{+}1)^1$ (successor), $\Pi^{\rho\to\tau\to\rho}$ and $\Sigma^{(\delta\to\rho\to\tau)\to(\delta\to\rho)\to\delta\to\tau}$ (combinators), and $\mathrm{R}^{0\to\rho\to(\rho\to0\to\rho)\to\rho}$ (recursor of type ρ). There is also the function symbol of juxtaposition which is not explicitly denoted: for terms $t^{\sigma\to\tau}$ and s^{σ}, $t\,s$ is a term of type τ.

The axioms defining these symbols are (the universal closures of each of):

$$x = x, \qquad x = y \to y = x, \qquad x = y \to y = z \to x = z, \qquad x = y \to x + 1 = y + 1,$$

$$x = y \to t[x/z] = t[y/z] \qquad \text{where } t[x/z] \text{ is the simultaneous substitution of } x \text{ for } z \text{ in } t$$

$$\begin{aligned} t[\Pi xy/u] &= t[x/u] \\ t[\Sigma xyz/u] &= t[xz(yz)/u] \\ t[\mathrm{R}0yz/u] &= t[y/u] \\ t[\mathrm{R}(x+1)yz/u] &= t[z(\mathrm{R}xyz)x/u] \end{aligned}$$

We also add the axiom schema of induction, for arbitrary formula $A(x)$, but only for variables x of type 0:

$$A(0) \to \forall x^0(A(x) \to A(x+1)) \to \forall x^0(A(x)) \tag{IA}$$

Since "=" is the only predicate symbol, all atomic (prime) formulas are of form $t = s$. This allows us to show that $x = y \to A(x) \to A(y)$, by induction on the complexity of formula A.

It is known that using the combinators one may define an individual term for lambda abstraction, denoted $\dot{\lambda}x.t$, of type 1, which satisfies the usual β-reduction axiom,

$$(\dot{\lambda}x^0.s^0)t^0 = s[t/x].$$

Using this and the recursor R, one can easily define all the usual primitive recursive functions. Using the thus defined predecessor function, and the induction axiom, one can derive the remaining Peano axioms, $x + 1 = y + 1 \to x = y$, and $(x + 1 = 0) \to 1 = 0$, where we took $1 = 0$ instead of $\bot$ because we are in minimal logic. In fact, in the presence of arithmetic, one can prove, again by induction, that the rule of $\bot$-elimination (with $\bot$ replaced by $1 = 0$) is derivable, although we will not need it.

Some notational conventions follow. We shall need to speak of bits, finite sequences of bits (bit-strings), and infinite sequences of bits (bit-streams). Bits and bit-strings can be encoded by natural numbers, but, instead of using the type 0 for terms of that kind, to be more pragmatic, we will write bool (intended to interpret $\mathbb{B}$) and bool^* (intended to interpret $\mathbb{B}^*$). Bitstreams are represented by terms of type $0 \to 0$, but we will write $0 \to \mathsf{bool}$ instead. We will need the operations for concatenation and initial segments of both bit-strings and bit-streams, that we already introduced. In addition, the operator $\mathrm{head}(p)$ returns the first bit of p, while $\mathrm{tail}(p)$ returns the string that follows the first bit of p. Although p is not a function, we will use the notation $p(n)$ to extract the $(n + 1)$-th bit of p[10]. We will also use the fact that one can define by primitive recursion a term $\mathsf{if}\ \cdots\ \mathsf{then}\ \cdots\ \mathsf{else}\ \cdots$ of type $\mathsf{bool} \to \mathsf{bool} \to \mathsf{bool} \to \mathsf{bool}$, such that the following equations hold:

$$\begin{aligned} \mathsf{if}\ 0\ \mathsf{then}\ y\ \mathsf{else}\ z &= z \\ \mathsf{if}\ x+1\ \mathsf{then}\ y\ \mathsf{else}\ z &= y \end{aligned}$$

We will also need the usual operation $\min : 0 \to 0 \to 0$ on numbers. All the mentioned operations can be defined by a restricted amount of primitive recursion at higher types, level 3 of the Grzegorcyk hierarchy would suffice. Hence we could work in a corresponding subsystem of HA^ω, like for example $\mathrm{G_3A}_i^\omega$ from §3.5 of [12].

Finally, we shall also need the following choice axiom, a restriction of the usual Axiom of Countable Choice ($\mathrm{AC}^{0,0}$):

$$\forall x^0 \exists! y^{\mathsf{bool}} A(x, y) \to \exists \phi^{0 \to \mathsf{bool}} \forall x^0 A(x, \phi\, x) \qquad (\mathrm{AC!}^{0,\mathbb{B}})$$

Neither $\mathrm{AC}^{0,0}$ nor $\mathrm{AC!}^{0,\mathbb{B}}$ is provable in HA^ω. For arithmetical formulas, $\mathrm{AC}^{0,0}$ (and hence $\mathrm{AC!}^{0,\mathbb{B}}$) is an admissible rule for HA^ω [2].

4.2 Proof term for OI-$\mathbb{B}$

We now formalize the concepts involved in the proof of OI-$\mathbb{B}$. An open set A in Cantor space is given, as a parameter to the logical system, by a term π of type $0 \to \mathsf{bool}^*$, an enumeration of basic opens. Each bit-string $\pi(n)$ is a basic open and the union of them

[10] $\mathrm{head}\, p$ (resp. $p(n)$) returns an arbitrary default value when p is an empty sequence (resp. $\mathrm{len}(p) < n+1$). However, we will use these operations only in a well-defined way.

makes A. Membership in A, $\alpha \in A$, means that α is covered by some basic open from the enumeration. Formally, we define

$$\alpha \in A \text{ iff } \exists l^0 \exists m^0 (\overline{\alpha}\, l = \pi(m)),$$

and we see that membership in A is a closed Σ-formula. (Recall that π is a parameter of the logical system.) The relation $<$ on bit-streams is formalized as

$$\beta < \alpha \text{ iff } \exists n^0 \left(\overline{\beta} n = \overline{\alpha} n \wedge (\beta(n) = 0 \wedge \alpha(n) = 1)\right).$$

We use an instance of Axiom 4 for the enumerable set B given by a Σ-formula $B(x)$, to be defined below, and n given by the natural number encoding an empty sequence. We define

$$B(x) := \exists k^0 \forall q^{\mathsf{bool}^k} \exists l^0 \exists m^0 (\overline{x * q}\, l = \pi(m)),$$

where $\forall q^{\mathsf{bool}^k}$ denotes a *bounded* universal quantification over bit-strings of length k. Bounded quantification can be encoded away using primitive recursive symbols, hence $B(x)$ is still a Σ-formula. We define $p \in B$ by $B(p)$. We have that, for any α, $\exists n(\overline{\alpha} n \in B)$ iff $\alpha \in A$. We instantiate the parameter S of $\mathrm{HA}^\omega_+(S)$ by $\langle\rangle \in B$.

Next, we give an interpretation of the instance of Axiom 4 in $\mathrm{HA}^\omega_+(\langle\rangle \in B)$. We cannot literally formalize Axiom 4 in $\mathrm{HA}^\omega_+(S)$, since $\mathrm{HA}^\omega_+(S)$ does not have higher-order quantification (but only quantification over higher types), hence we cannot quantify over subsets. We therefore "interpret" (the instance of) Axiom 4:

$$\forall s^{\mathsf{bool}^*} \Big(\forall \chi_C^{\mathsf{bool}^* \to \mathsf{bool}} \Big(\forall x^{\mathsf{bool}^*} (\chi_C(x) = 1 \to B(x)) \to \\ \exists q^{\mathsf{bool}^*} (\chi_C(q) = 1 \to B(s)) \to \ \exists r^{\mathsf{bool}^*} ((\chi_C(r) = 1 \to B(s)) \wedge B(r))\Big)\Big) \to B(\langle\rangle).$$

The enumerable set B is represented by the Σ-formula $B(x)$, the decidable subset C by a characteristic function $\chi_C^{\mathsf{bool}^* \to \mathsf{bool}}$, replacing the premise $\forall x\, (x \in C \vee (x \in C \to s \in B))$. The characteristic function should intuitively read as $\chi_C(p) = 1$ iff "$p \in C$", but we take $B(s)$ for $\bot$.

The proof term for OI-$\mathbb{B}$ is shown in Figure 1. We obtained it by formalizing the proofs of Theorems 1 and 4 in $\mathrm{HA}^\omega_+(\langle\rangle \in B)$, and then by normalizing and (hand-)optimizing the formalized proof term, to obtain a compact and direct program proving OI-$\mathbb{B}$.

To ease the presentation, at certain places, we have put after a semicolon the type annotations for individual terms, and the formulas for proof terms. Some parts, being too long, have been put below the main proof term. We suppress the use of equality axioms, to keep the proof term simple without equality-rewriting terms. It is known that equality proofs have no computational content when extracting programs, as they are realized by singleton data types.

We now explain the behavior of the proof term. Given a proof h that A is progressive, it has to show that $\alpha' \in A$ for any α'. As in the proof of Theorem 1, it proves $\langle\rangle \in B$ (lines 3-10), from which we obtain k' such that $h^5 : \forall q^{\mathsf{bool}^{k'}} \exists l^0 \exists m^0 (\overline{q}\, l = \pi(m))$ (line 10). Then $h^5(\overline{\alpha'} k')$ gives us j' such that $h^6 : \exists m^0 (\overline{\overline{\alpha'} k' j'} = \pi(m))$ (line 11), so that $(\min(k', j'), h^6)$ proves $\exists l^0 \exists m^0 (\overline{\alpha'} l = \pi(m))$ (line 12). (An explicit proof of the equality $\overline{\overline{\alpha'} k' j'} = \overline{\alpha'}(\min(k', j'))$ would need an explicit definition of the min function and induction).

To show $\langle\rangle \in B$, which is the parameter of the system, it applies a reset # (line 3), and now it has to show the same formula, but classical logic in the form of the shift rule

$$
\begin{array}{ll}
1: & \lambda h : \forall\alpha(\forall\beta < \alpha(\beta \in A) \to \alpha \in A).\tilde{\lambda}\alpha'. \\
2: & \mathsf{dest} \\
3: & \quad \Big(\#\mathsf{dest}\ a_C(\tilde{\lambda}x.\mathcal{S}k.k(\iota_2(\lambda a.k(\iota_1 a))))\ \mathsf{as}\ (\chi.b)\ \mathsf{in} \\
4: & \qquad \mathsf{dest}\ \Big(h\alpha\big(\tilde{\lambda}\beta.\lambda h' : \beta < \alpha. \\
5: & \qquad\quad \mathsf{dest}\ (h' : \beta < \alpha)\ \mathsf{as}\ (n.h'')\ \mathsf{in} \\
6: & \qquad\quad \mathsf{dest}\ (a_1(\pi_2\pi_2 h'') : \overline{\beta}(n+1) \in B)\ \mathsf{as}\ (k.h''')\ \mathsf{in} \\
7: & \qquad\quad \mathsf{dest}\ (h'''(\langle\beta(n+1)\rangle * \cdots * \langle\beta(n+k)\rangle) : \overline{\beta}(n+k+1) \in A)\ \mathsf{as}\ (j.h^4)\ \mathsf{in} \\
8: & \qquad\quad (\min(n+k+1, j), h^4)\big) : \alpha \in A\Big)\ \mathsf{as}\ (l.c)\ \mathsf{in} \\
9: & \qquad \mathsf{dest}\ (c : \exists m(\overline{\alpha}l = \pi(m)))\ \mathsf{as}\ (m.d)\ \mathsf{in} \\
10: & \qquad a_I\,(\lambda h.h)\,a_3\,l\,(0, \tilde{\lambda}q.(l,(m,d))) : \langle\rangle \in B\Big)\ \mathsf{as}\ (k'.h^5)\ \mathsf{in} \\
11: & \quad \mathsf{dest}\ (h^5\,(\overline{\alpha'}k') : \overline{\alpha'}k' \in A)\ \mathsf{as}\ (j'.h^6)\ \mathsf{in} \\
12: & \quad (\min(k', j'), h^6)
\end{array}
$$

$$
\begin{array}{l}
\alpha := \dot{\lambda}n. \\
\quad \mathrm{R}(n+1, \langle\rangle, (\dot{\lambda}z.\dot{\lambda}n'.z * \langle \mathsf{if}\ \chi(z * \langle 0\rangle)\ \mathsf{then}\ (\mathsf{if}\ \chi(z * \langle 1\rangle)\ \mathsf{then}\ 0\ \mathsf{else}\ 1)\ \mathsf{else}\ 0\rangle))(n) \\
a_1 : \alpha(n) = 1 \to \overline{\beta}(n+1) \in B := \lambda h.\mathsf{case}\ a_B(\chi(\overline{\beta}(n+1)))\ \mathsf{of} \\
\quad \big(h_1.(\pi_1(b(\overline{\beta}(n+1))))\,h_1 \| h_2.(\pi_1(b(\overline{\beta}(n+1))))\,h_2\big) \\
a_3 := \tilde{\lambda}n.\lambda h_I : \overline{\alpha}n \in B \to \langle\rangle \in B.\lambda h : \overline{\alpha}(n+1) \in B. \\
\quad \mathsf{case}\ a_B(\chi(\overline{\alpha}n * \langle 0\rangle))\ \mathsf{of}\ (h_1.(\pi_2(b(\overline{\alpha}(n+1))))\,h_1\,h \\
\qquad \| h_2.\mathsf{case}\ (a_B(\chi(\overline{\alpha}n * \langle 1\rangle)))\ \mathsf{of}\ \ (h_{21}.(\pi_2(b(\overline{\alpha}(n+1))))\,h_{21}\,h \| h_{22}.h_I\,a_4)) \\
a_4 : \overline{\alpha}n \in B := \\
\quad \mathsf{dest}\ ((\pi_1(b(\overline{\alpha}n * \langle 0\rangle)))\,h_2 : \overline{\alpha}n * \langle 0\rangle \in B) \\
\qquad \mathsf{as}\ (k_0.f_0 : \forall q : \mathsf{bool}^{k_0}.\exists l, m(\overline{\overline{\alpha}n * \langle 0\rangle * q}\ l = \pi(m)))\ \mathsf{in} \\
\quad \mathsf{dest}\ ((\pi_1(b(\overline{\alpha}n * \langle 1\rangle)))\,h_{22} : \overline{\alpha}n * \langle 1\rangle \in B) \\
\qquad \mathsf{as}\ (k_1.f_1 : \forall q : \mathsf{bool}^{k_1}.\exists l, m(\overline{\overline{\alpha}n * \langle 1\rangle * q}\ l = \pi(m)))\ \mathsf{in} \\
\quad (\min(k_0, k_1) + 1, \lambda q : \mathsf{bool}^{\min(k_0,k_1)+1}.\mathsf{if}\ \ \mathrm{head}(q)\ \mathsf{then}\ f_1(\overline{\mathrm{tail}(q)}k_1)\ \mathsf{else}\ f_0(\overline{\mathrm{tail}(q)}k_0))
\end{array}
$$

Figure 1 Proof term for OI-$\mathbb{B}$ of type $((\forall\alpha(\forall\beta < \alpha(\beta \in A) \to \alpha \in A)) \to \forall\alpha'(\alpha' \in A))$ in $\mathrm{HA}^{\omega}_{+}(\langle\rangle \in B)$.

can be used. Indeed, the proof term $\tilde{\lambda}x.\mathcal{S}k.k(\iota_2(\lambda a.k(\iota_1 a)))$ proves the "decidability" of B: $\forall x^{\mathsf{bool}^*}(x \in B \vee (x \in B \to \langle\rangle \in B))$. Using the proof term a_C for the formula

$$
\begin{array}{l}
\forall x^{\mathsf{bool}^*}(x \in B \vee (x \in B \to \langle\rangle \in B)) \to \\
\qquad \exists\chi^{\mathsf{bool}^* \to \mathsf{bool}}\forall x^{\mathsf{bool}^*}((\chi(x) = 1 \to x \in B) \wedge (\chi(x) = 0 \to (x \in B \to \langle\rangle \in B))),
\end{array}
$$

we obtain from the decidability, a characteristic function $\chi^{\mathsf{bool}^* \to \mathsf{bool}}$ for B. The proof term a_C is constructed by combining AC!$^{0,\mathbb{B}}$ together with a proof term that eliminates disjunction in presence of arithmetic[11]. The proof term b proves the characteristic property of χ, namely, $\forall x((\chi(x) = 1 \to x \in B) \wedge (\chi(x) = 0 \to (x \in B \to \langle\rangle \in B)))$.

[11] For the proof of this statement, $(A \vee B) \leftrightarrow \exists x((x = 1 \to A) \wedge (x = 0 \to B))$, see for example §§1.3.7 of [17].

Now, using this χ, the bit-stream α that we saw in the proof of Theorem 1 can be constructed using R and $\mathsf{if}\ \cdots\ \mathsf{then}\ \cdots\ \mathsf{else}\ \cdots$ by (encoded) course-of-values recursion.

Next one needs to show that $\alpha \in A$ (lines 4-8). One uses progressiveness h: from β and a proof h' of $\beta < \alpha$, one extracts n and a proof h'' of

$$\overline{\beta}n = \overline{\alpha}n \wedge (\beta(n) = 0 \wedge \alpha(n) = 1).$$

Then, $\pi_2\pi_2 h''$ shows $\alpha(n) = 1$, and it is for a_1 to show that $\overline{\alpha}n * \langle 0\rangle = \overline{\beta}(n+1)$ is in B, which in turn shows, with the help of h''', that $\overline{\beta}(n+k+1) \in A$, i.e., $\exists j \exists i(\overline{\beta}(n+k+1)j = \pi(i))$[12]. Now, one concludes $\beta \in A$ with $(\min(n+k+1, j), h^4)$ by appropriately choosing the witness $\min(n+k+1, j)$ so that $\overline{\beta}(n+k+1)j = \overline{\beta}(\min(n+k+1, j))$ holds. (Again, we suppress the proof term for this equality.)

The proof term a_1 derives $\overline{\beta}(n+1) \in B$ from $\alpha(n) = 1$ by making a case distinction. To generate the disjunction needed for the case analysis, one uses a proof term a_B for $\forall x^{\mathsf{bool}}(x = 0 \vee x = 1)$. For the first case in which $\chi(\overline{\beta}(n+1)) = 0$, we have an absurdity $1 = 0$, by definition of α, since $\alpha(n) = 1$. Hence, by equality-rewriting we may use the proof term h_1 at type $\chi(\overline{\beta}(n+1)) = 1$. Now, both the two cases are closed by applying $\pi_1(b(\overline{\beta}(n+1)))$, which proves $\chi(\overline{\beta}(n+1)) = 1 \to \overline{\beta}(n+1) \in B$, to h_1 and h_2, respectively.

From $\alpha \in A$, one obtains the length l and the index m such that $\overline{\alpha}l$ is covered by the basic open $\pi(m)$ (the proof term d in line 9), and then one can show that $\overline{\alpha}0 = \langle\rangle$ is in B. This last fact is derived by the proof term

$$a_I\,(\lambda h.h)\,a_3\,l\,(0, \tilde{\lambda}q.(l, (m, d))),$$

where a_I is a proof term behind an instance of the induction axiom showing $\forall l^0(\overline{\alpha}l \in B \to \langle\rangle \in B)$. The proof term a_I uses the proof term a_3 which derives

$$\forall n((\overline{\alpha}\,n \in B \to \langle\rangle \in B) \to \overline{\alpha}\,(n+1) \in B \to \langle\rangle \in B).$$

It is proved by case analysis, considering the possibilities for the pair $(\chi(\overline{\alpha}n * \langle 0\rangle), \chi(\overline{\alpha}n * \langle 1\rangle))$. If either $\chi(\overline{\alpha}n * \langle 0\rangle) = 0$ or $\chi(\overline{\alpha}n * \langle 1\rangle) = 0$ holds, we close the case by the characteristic property of χ together with the hypothesis h. Otherwise, i.e. both $\chi(\overline{\alpha}n * \langle 0\rangle) = 1$ and $\chi(\overline{\alpha}n * \langle 1\rangle) = 1$ holds, we can deduce $\overline{\alpha}n \in B$ (the proof term a_4), from which the case follows by the induction hypothesis.

5 Conclusion

We gave a direct proof for OI-$\mathbb{B}$ in a constructive predicate logic incorporating delimited control operators. While computational interpretation of $\mathrm{MQC}_+(S)$ is available, namely the standard call-by-value weak-head reduction semantics for lambda calculus with shift and reset, we cannot directly analyze the computational behavior of the proof term for OI-$\mathbb{B}$ because, at the moment, we do not have a proof term for $\mathrm{AC}!^{0,\mathbb{B}}$ used in the proof term for OI-$\mathbb{B}$. The best way to overcome this limitation would be to extend $\mathrm{MQC}_+(S)$ so that it can derive $\mathrm{AC}!^{0,\mathbb{B}}$ as it is done in Martin-Löf Type Theory or constructive versions of Hilbert's epsilon calculus.

Another way to overcome the limitation would be to use a realizability or functional interpretation that extracts programs from constructive proofs even in presence of choice

[12] The proof term $a_1(\pi_2\pi_2 h'')$ proves $\overline{\alpha}n * \langle 0\rangle \in B$, from which $\overline{\beta}(n+1) \in B$ follows using equality axioms. As remarked earlier, equality-rewriting is implicit in the proof term.

axioms. For example, by using Spector's extension of Gödel's functional interpretation with bar recursion, we could extract a program from our proof. However, to replace bar recursion is the point of using delimited control operators in the first place.

If and when our future work is successful, it would allow, at least for the case of the compact Cantor space, to replace Berger's general-recursive computation schema of *open recursion* by a terminating computation schema based on control operators.

The work of Krivine on Classical Realizability gives an interpretation of the Axiom of Dependent Choice [13] using control operators for classical logic. Herbelin recently gave a more direct version of that work [10], using classical control operators and coinduction.

Finally, we would like to mention Veldman's recent work in Constructive Reverse Mathematics [19, 20] that has served as inspiration for our work. An article of Veldman on the equivalence of Open Induction with a number of other axioms is in preparation. In our paper, we showed one direction of this equivalence for the topology of Cantor space seen as the infinite binary tree rather than as the subset of the real line.

Acknowledgments. We would like to thank Wim Veldman for explaining us some of his results, and Ralph Matthes and Hugo Herbelin for valuable comments on the draft.

References

1 Zena M. Ariola and Hugo Herbelin. Minimal classical logic and control operators. In *Thirtieth International Colloquium on Automata, Languages and Programming, ICALP'03, Eindhoven, The Netherlands, June 30 to July 4, 2003*, volume 2719 of *Lecture Notes in Computer Science*, pages 871–885. Springer, 2003.

2 Michael Beeson. Goodman's theorem and beyond. *Pacific Journal of Mathematics*, 84:1–16, 1979.

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

4 Thierry Coquand. Constructive topology and combinatorics. In J. Myers and M. O'Donnell, editors, *Constructivity in Computer Science*, volume 613 of *Lecture Notes in Computer Science*, pages 159–164. Springer Berlin / Heidelberg, 1992. DOI: 10.1007/BFb0021089.

5 Thierry Coquand. A note on the open induction principle, 1997.

6 Olivier Danvy and Andrzej Filinski. A functional abstraction of typed contexts. Technical report, Computer Science Department, University of Copenhagen, 1989. DIKU Rapport 89/12.

7 Olivier Danvy and Andrzej Filinski. Abstracting control. In *LISP and Functional Programming*, pages 151–160, 1990.

8 Olivier Danvy and Andrzej Filinski. Representing control: A study of the CPS transformation. *Mathematical Structures in Computer Science*, 2(4):361–391, 1992.

9 Hugo Herbelin. An intuitionistic logic that proves Markov's principle. In *Proceedings, 25th Annual IEEE Symposium on Logic in Computer Science (LICS'10), Edinburgh, UK, 11–14 July 2010*, page N/A. IEEE Computer Society Press, 2010.

10 Hugo Herbelin. A constructive proof of dependent choice, compatible with classical logic. In *Proceedings of the 27th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2012, 25–28 June 2012, Dubrovnik, Croatia*, pages 365–374. IEEE Computer Society, 2012.

11 Danko Ilik. Delimited control operators prove double-negation shift. *Annals of Pure and Applied Logic*, 163(11):1549–1559, 2012.

12 Ulrich Kohlenbach. *Applied proof theory: proof interpretations and their use in mathematics*. Springer Monographs in Mathematics. Springer-Verlag, Berlin, 2008.

13 Jean-Louis Krivine. Dependent choice, 'quote' and the clock. *Theor. Comput. Sci.*, 308(1–3):259–276, 2003.

14 Chetan Murthy. *Extracting Classical Content from Classical Proofs*. PhD thesis, Department of Computer Science, Cornell University, 1990.

15 Jean-Claude Raoult. Proving open properties by induction. *Information Processing Letters*, 29:19–23, 1988.

16 Clifford Spector. Provably recursive functionals of analysis: a consistency proof of analysis by an extension of principles formulated in current intuitionistic mathematics. In *Proc. Sympos. Pure Math., Vol. V*, pages 1–27. American Mathematical Society, Providence, R.I., 1962.

17 Anne S. Troelstra, editor. *Metamathematical Investigations of Intuitionistic Arithmetic and analysis*. Lecture Notes in Mathematics 344. Springer-Verlag, 1973.

18 Wim Veldman. The principle of open induction on the unit interval [0,1] and some of its equivalents. Slides from presentation, May 2010.

19 Wim Veldman. Brouwer's Fan Theorem as an axiom and as a contrast to Kleene's Alternative. *ArXiv e-prints*, June 2011.

20 Wim Veldman. Some further equivalents of Brouwer's Fan Theorem and of Kleene's Alternative. *ArXiv e-prints*, November 2013.