Effective Operations of Type 2 in Pcas

Eric Faber

Jaap van Oosten
Department of Mathematics, Utrecht University
P.O. Box 80.010, 3508TA Utrecht
The Netherlands
J.vanOosten@uu.nl

August 19, 2014

Abstract

We exhibit a way of "forcing a functional to be an effective operation" for arbitrary partial combinatory algebras (pcas). This gives a method of defining new pcas from old ones for some fixed functional, where the new partial functions can be viewed as computable relative to that functional. It is shown that this generalizes a notion of computation relative to a functional as defined by Kleene for the natural numbers.

The construction can be used to study subtoposes of the Effective Topos. We will do this for a particular functional that forces every arithmetical set to be decidable.

In this paper we also prove the convenient result that the two definitions of a pca that are common in the literature are essentially the same.

1 Introduction

Effective operations have originally been studied in computability theory, with as famous results the theorems of Myhill-Shepherdson and Rice-Shapiro. They have also been studied in relation to computable functionals (e.g. see [3]). For example, it is known that not every functional that is an effective operation is a computable functional. In realizability, partial combinatory algebras (pcas) are used as models for abstract Turing machines. The notion of effective operation easily extends to pcas. In this paper, we discuss effective operations on arbitrary partial combinatory algebras, and present a way to force a functional to be an effective operation: for a (total) functional $F : A^A \to A$, where A is a pca, we construct a pca $A[F]$ for which F is an effective operation.

In [8], Kleene introduced the notion of a *computable functional*, and with it the notion of computability relative to a functional. It turns out that for $A = \mathcal{K}_1$, Kleene's first model, the functions computable in $\mathcal{K}_1[F]$ are precisely the functions computable relative to F in Kleene's sense. This yields an alternative approach to computation in a type 2 functional that generalizes to arbitrary pcas. We can use this to study local operators in realizability toposes. The theory of recursive hierarchies of functionals (e.g. see [4]) can therefore be applied to study subtoposes of realizability toposes.

As an example, we look at a specific local operator J that is studied in [15]. It turns out that the functions "computable in J" are precisely the functions computable relative to a certain functional. On the way, we also prove that a strengthening of our definition of a pca, that we call strict pcas, is (up to isomorphism) not a proper strengthening at all. This is posed as an open question in [2]. The result is convenient, since in the literature there has not been wide consensus on the definition of pca; several authors have been using strict pcas by default.

2 Preliminaries

We start by reviewing the definition of a partial combinatory algebra and some basic constructions.

Definition 2.1. A partial applicative structure on a set A is a partial map $A \times A \to A$ that we denote by

$$(a, b) \mapsto ab.$$

We write $ab\downarrow$ to say that ab is defined, $ab = c$ to say that ab is defined and has value c. In writing compositions of the application, we adopt the convention of *associating to the left*, so:

$$abcd = (((ab)c)d).$$

For $V = \{x_0, x_1, \ldots\}$ an enumerable set of variables, we define the set of *terms* by:

Definition 2.2. For all $a \in A$, $x \in V$

1. a is a term
2. x is a term
3. For t, s terms, (ts) is a term.

A term t without free variables is called *closed* and might *denote* or not. This is defined inductively:

1. a denotes and has value a
2. If t denotes and has value a, and s denotes and has value b, and $ab = c$, then (ts) denotes and has value c.

We write $t\downarrow$ if t denotes, and $t = a$ if t denotes and has value a. We write $t \lesssim s$ if $s\downarrow$ implies that $t\downarrow$, and in that case $s = t$. We write $t \simeq s$ if $t \lesssim s$ and $s \lesssim t$.

We denote a term t with free variables $x_1, \dots, x_n$ by

$$t(x_1, \dots, x_n)$$

. For $a_1, \dots, a_n \in A$, we can substitute $x_1, \dots, x_n$ by $a_1, \dots, a_n$, resulting in a closed term $t(a_1, \dots, a_n)$.

Definition 2.3. A partial combinatory algebra is a partial applicative structure A, with elements $\mathsf{k}, \mathsf{s} \in A$ such that for all $a, b \in A$:

1. $\mathsf{s}ab\downarrow$
2. $\mathsf{k}ab = a$
3. $\mathsf{s}abc \lesssim ac(bc)$.

We also say that the applicative structure on A is *combinatory complete*.

Example 2.4. There are several examples of pcas, [14] treats a lot of them. Here we just list a few with little explanation:

- The trivial pca, $A = \{*\}$ with $** = *$.
- *Kleene's first model* $\mathcal{K}_1$, with underlying set $\mathbb{N}$. Application is given by:

  $$nm \simeq \phi_n(m)$$

  where ϕ_n is the $n-$th partial function for some model of computation and coding.

An introduction to partial combinatory algebras and its basic properties can be found in chapter 1 of [14]. Here we will just quickly present some tools that we need later on, the proofs can all be found in [14]. For a pca A, an element $a \in A$ defines a partial function $A \to A$ by:

$$b \mapsto ab.$$

We say that a is an *index* for this partial function. By abuse of language, we sometimes say that a is a partial (computable) function.

An important result for pcas is the *recursion theorem*:

Theorem 2.5 (Recursion Theorem). *For every $f \in A$, there exists e such that for all $a \in A$:*

$$ea \lesssim fea.$$

Also important is the fact that a pca is *combinatory complete*. This means that for every term $t(x_1, \dots, x_n)$ with free variables $x_1, \dots, x_n$, there is an element $\langle x_1 \cdots x_n \rangle t \in A$ such that for all $a_1, \dots, a_n \in A$:

$$(\langle x_1 \cdots x_n \rangle t)a_1 \cdots a_{n-1}\downarrow \tag{1}$$

$$(\langle x_1 \cdots x_n \rangle t)a_1 \cdots a_{n-1}a_n \lesssim t(a_1, \dots, a_n). \tag{2}$$

The notation $\langle x_1 \cdots x_n \rangle t$ should remind the reader of $\lambda-$abstraction in $\lambda-$calculus. We use a different notation because it is slightly different from $\lambda-$abstraction, but it is applied in the same way.

Besides k and s, we identify (a choice of) other canonical elements of A, among which:

- Boolean combinators T and F that satisfy for all $a, b \in A$:

$$\mathsf{T}ab = a \tag{3}$$
$$\mathsf{F}ab = b. \tag{4}$$

  Indeed, one can take $\mathsf{T} = \mathsf{k}$.

- Using combinatory completeness and the boolean combinators, we can define if/else statements: for t, s closed terms:

$$(\langle v \rangle v(\langle x \rangle t)(\langle x \rangle s)\mathsf{k})\, b \lesssim \begin{cases} t & \text{if } b = \mathsf{T} \\ s & \text{if } b = \mathsf{F} \\ \text{unspecified} & \text{otherwise.} \end{cases}$$

  For legibility, we denote a term like $v(\langle x \rangle t)(\langle x \rangle s)\mathsf{k}$ by

$$\mathsf{if}\ v\ \mathsf{then}\ t\ \mathsf{else}\ s.$$

- The pairing combinator p together with projections $\mathsf{p}_0, \mathsf{p}_1$:

$$\mathsf{p}_0(\mathsf{p}ab) = a \tag{5}$$
$$\mathsf{p}_1(\mathsf{p}ab) = b \tag{6}$$

- The Curry numerals: each pca contains representables of the natural numbers, i.e. for any $n \in \mathbb{N}$ an element $\overline{n} \in A$. It is a basic result (for a proof, see [14]) that, for every k, there is for every $k-$ary partial recursive function $F : \mathbb{N}^k \to \mathbb{N}$ an index $f \in A$ such that for all $(n_1, \ldots, n_k) \in \operatorname{dom} F$:

$$f\overline{n_1} \ldots \overline{n_k} = \overline{F(n_1, \ldots, n_k)}.$$

When there is chance of confusion, we decorate (choices of) canonical elements by a subscript to denote the pca to which it belongs, e.g. $\mathsf{k}_A, \mathsf{s}_A, \mathsf{T}_A$,etc.

We can also code tuples of elements in a computable way. That is, there is an index $t \in A$ such that for all $n \in \mathbb{N}$, $u_0, \ldots, u_n \in A$ we can code a tuple

$$[u_0, \ldots, u_n] := t\overline{n+1}u_0 \cdots u_n \in A$$

such that projection, concatenation of tuples, and a length function (that yields the length of a tuple as a curry numeral) are all computable. We denote concatenation by $*$:

$$[u_0 \ldots, u_i] * [u_{i+1}, \ldots, u_n] = [u_0, \ldots, u_n]$$

and for $u = [u_0, \ldots, u_n]$, we write $u^{<i+1}$ for $[u_0, \ldots, u_i]$. These basic operations can all be done computably and uniformly in the variables.

There is a notion of morphism between partial combinatory algebras due to John Longley (see [9]), called *applicative morphism*. In the following definition, we adopt the following notation: for a total relation $R : X \to Y$ between sets X and Y, we denote for every $x \in X$ by $R(x)$ the set:

$$\{y \in Y \mid xRy\}.$$

For A a pca, $a \in A$ and $P \subseteq A$ a subset, we write $aP\downarrow$ if for all $b \in P$, $ab\downarrow$, and in that case

$$aP := \{ab \mid b \in P\}.$$

For $P, Q \subseteq A$, we write $PQ\downarrow$ if for all $b \in P, b' \in Q$, $bb'\downarrow$, and in that case

$$PQ = \{bb' \mid b \in P, b' \in Q\}.$$

Whenever we write compositions of these applications of an element and a subset, or a subset and a subset, we again associate to the left.

Definition 2.6 (Longley)**.** Let A, B be pcas. An applicative morphism $\gamma : A \to B$ is a total relation from A to B, together with an element $r \in B$ such that whenever $a, a' \in A$ and $aa'\downarrow$, we have:

$$r\gamma(a)\gamma(a')\downarrow \text{ and } r\gamma(a)\gamma(a') \subseteq \gamma(aa').$$

The element r is said to *realize* γ.

An applicative morphism γ is *decidable* if there is $d \in B$ such that

$$d\gamma(\mathsf{T}_A) = \{\mathsf{T}_B\} \tag{7}$$

$$d\gamma(\mathsf{F}_A) = \{\mathsf{F}_B\}. \tag{8}$$

We call d a *decider* for γ.

The composition of two applicative morphisms is defined by composition of the relations, and one can verify that this is again an applicative morphism. There is a pre-order on applicative morphisms defined as follows: for $\gamma : A \to B$, $\delta : A \to B$, write $\gamma \preceq \delta$ when there is $t \in B$ such that for all $a \in A$:

$$t\gamma(a) \subseteq \delta(a).$$

We write $\gamma \simeq \delta$ if $\gamma \preceq \delta$ and $\delta \preceq \gamma$, in this case γ and δ are *equivalent*. One can show that this pre-order is preserved by composition on both sides. We thus obtain a pre-order enriched category PCA consisting of partial combinatory algebras as objects, and applicative morphisms as arrows. The identity arrow for a pca A is denoted by ι_A and given by the identity relation on A.

Note. As remarked in the introduction, definition 2.3 is weaker than the definition that is used in some other literature (including [14]). However, it will be a consequence of theorem 5.8 below that the corresponding categories of pcas, with morphisms as above, are equivalent.

For a pca A, we define a set PR_A by:

$$\mathrm{PR}_A = \{f : A \to A \text{ partial } \mid (\exists a \in A)(\forall b \in A) ab \precsim f(b)\}.$$

We also define a function $I_1^A : \mathrm{PR}_A \to A$ by:

$$I_1^A(f) = \{a \in A \,|\, (\forall b \in A) ab \precsim f(b)\}.$$

For $\gamma : A \to B$ an applicative morphism, we define a set PR_γ by:

$$\mathrm{PR}_\gamma : \{f : A \to A \text{ partial } \mid (\exists b \in B)(\forall a \in A) f(a)\downarrow \Rightarrow b\gamma(a) \subseteq \gamma(f(a))\}. \quad (9)$$

Lastly, we also define a function $I_1^\gamma : \mathrm{PR}_\gamma \to B$ by:

$$I_1^\gamma(f) = \{b \in B \,|\, (\forall a \in A) f(a)\downarrow \Rightarrow b\gamma(a) \subseteq \gamma(f(a))\}.$$

Observe that $\mathrm{PR}_A = \mathrm{PR}_{\iota_A}$.

The following result by the second author (see [13] or [14]) tells us that we can adjoin any partial function $f : A \to A$ to a pca A, to obtain a pca $A[f]$ in which f has an index. This generalizes the notion of a Turing machine with oracle from ordinary computability theory to pcas.

Theorem 2.7 ([13])**.** *For A a pca and $f : A \to A$ a partial function, there exists a pca $A[f]$ defined on the set A such that the identity on A is a decidable applicative morphism $\iota_f : A \to A[f]$, and:*

1. *$f \in PR_{A[f]} = PR_{\iota_f}$.*
2. *Whenever $\gamma : A \to B$ is a decidable applicative morphism such that $f \in PR_\gamma$, there exists a decidable $\gamma_f : A[f] \to B$ such that $\gamma_f \circ \iota_f = \gamma$. Moreover, if $\delta : A[f] \to B$ is a decidable applicative morphism such that $\delta \circ \iota_f \cong \gamma$, then $\delta \cong \gamma_f$.*

The proof of the above theorem is done by defining an applicative structure on A using *dialogues*:

Definition 2.8. Let $f : A \to A$ be a partial function, and $a, b \in A$. An $f-$*dialogue* between a and b is an element $u = [u_0, \ldots, u_n] \in A$ such that for all $i \leq n$:

$$a([b] * u^{<i}) = \mathsf{pF} v_i$$

where $f(v_i)$ is defined and $f(v_i) = u_i$.

An $f-$dialogue u between a and b *halts* if there exists $c \in A$ such that

$$a([b] * u) = \mathsf{pT} c.$$

We write $a \cdot^f b\downarrow$ if there is a (necessarily unique) halting $f-$dialogue u between a and b, and $a \cdot^f b = c$ if $a \cdot^f b\downarrow$ and

$$a([b] * u) = \mathsf{pT} c$$

where u is a a halting $f-$dialogue between a and b.

The applicative structure on $A[f]$ is then given by $(a, b) \mapsto a \cdot^f b$. For a full proof of combinatory completeness, see [14] or [13]. Note that these sources use the strict version of a pca (definition 5.7), but the proofs work without any modification (alternatively, one can apply theorem 5.8). The following corollary corresponds to corollary 1.3 in [13]:

Corollary 2.9. *Let A be a pca.*

1. *If $f \in PR_A$, then $A[f]$ and A are isomorphic pcas.*
2. *If $f, g : A \to A$ are partial functions, then $A[f][g]$ and $A[g][f]$ are isomorphic.*
3. *If $\mathcal{K}_1^f$ is the pca of partial recursive application with an oracle for f, then $\mathcal{K}_1^f$ is isomorphic to $\mathcal{K}_1[f]$.*

3 Effective operations

We will now define the set of effective operations on pcas. For any pca A, let Tot_A be the subset of PR_A consisting of all total functions:

$$\mathrm{Tot}_A = \{f \in \mathrm{PR}_A \mid f \text{ is total }\}.$$

Similarly, define for any γ the subset $\mathrm{Tot}_\gamma \subseteq \mathrm{PR}_\gamma$ consisting of all total functions in PR_γ.

Definition 3.1 (Effective Operation)**.** For a pca A, we define the set E_A of *effective operations* in A by:

$$E_A = \{F : A^A \to A \text{ partial} \mid (\exists a \in A)(\forall f \in \mathrm{Tot}_A) F(f) \text{ defined } \Rightarrow \quad (10)$$

$$(aI_1(f)\downarrow \text{ and } aI_1^A(f) = \{F(f)\})\}. \quad (11)$$

Define a function $I_2^A : E_A \to A$ by:

$$I_2^A(F) = \{a \mid (\forall f \in \mathrm{Tot}_A)(\forall b \in I_1^A(f))(ab \precsim F(f))\}$$

Similarly, for $\gamma : A \to B$ an applicative morphism we define the set E_γ of *effective operations relative to* γ by:

$$E_\gamma = \{F : A^A \to A \text{ partial} \mid (\exists b \in B)(\forall f \in \mathrm{Tot}_\gamma) F(f) \text{ defined } \Rightarrow \quad (12)$$

$$(bI_1^\gamma(f)\downarrow \text{ and } bI_1^\gamma(f) \subseteq \gamma(F(f)))\}. \quad (13)$$

Example 3.2. As an example, we look at the following functional $E : A^A \to A$, where A is an arbitrary pca.

$$E(\alpha) = \begin{cases} \mathsf{T} & \text{if } (\exists a)\alpha(a) = \mathsf{T} \\ \mathsf{F} & \text{otherwise.} \end{cases}$$

For $A = \mathcal{K}_1$, it is easily seen that the above is not an effective operation (otherwise the halting problem is decidable). However, the below construction shows that there are a lot of pcas in which E is an effective operation.

The statement of the following theorem is very similar to theorem 2.7.

Theorem 3.3. *Let A be a pca, and $F : A^A \to A$ a (partial) functional. Then there exists a pca $A[F]$ defined on the set A such that the identity relation on A is a decidable applicative morphism $\iota_F : A \to A[F]$ and:*

1. *$F \in E_{A[F]} = E_{\iota_F}$*
2. *Whenever $\gamma : A \to B$ is a decidable applicative morphism such that $F \in E_\gamma$, there exists a unique (up to equivalence) decidable applicative morphism $\gamma_F : A[F] \to B$ such that $\gamma_F \circ \iota_F = \gamma$. Moreover, if $\delta : A[F] \to B$ is a decidable applicative morphism such that $\delta \circ \iota_F \cong \gamma$, then $\delta \cong \gamma_F$.*

Proof. We will construct $A[F]$ as $A[f]$ for

$$f = \bigcup_\alpha f_\alpha,$$

where $\{f_\alpha\}_\alpha$ is a compatible family of partial functions indexed by the ordinals. We define this family by transfinite induction. For all $a, b \in A$, we let:

- $f_0 = \emptyset$
- $f_{\alpha+1}(a) = b$ whenever there exists $g \in \mathrm{Tot}_{A[f_\alpha]}$ such that $a \in I_1^{A[f_\alpha]}(g)$ and $F(g) = b$.
- For $\lambda > 0$ a limit ordinal, $f_\lambda = \bigcup_{\alpha<\lambda} f_\alpha$.

It is not hard to see that this indeed defines a compatible family of partial functions. So $f = \bigcup_\alpha f_\alpha$ is a partial function, and we define $A[F] := A[f]$. We denote the application in $A[F]$ by $\cdot^F$.

For part (i), suppose $g \in \mathrm{Tot}_{A[F]}$, $a \in I_1^{A[F]}(g)$, and $F(g)$ is defined. Then it is not hard to see that there is α such that $g \in \mathrm{Tot}_{A[f_\alpha]}$, $a \in I_1^{A[f_\alpha]}(g)$. Then $f_{\alpha+1}(a) = F(g)$, hence $f(a) = F(g)$. Indeed, F is an effective operation in $A[F]$.

For part (ii): Suppose $\gamma : A \to B$ is a decidable applicative morphism such that $F \in E_\gamma$. Let $r_F \in I_2^\gamma(F)$. Denote the realizer for γ by r. Let d be the decider for γ. Let c be such that for all $x \in \gamma(a)$, $v \in \gamma(u)$, $\mathsf{c}xy \in \gamma([a] * u)$. Let c' be such that for all $y \in \gamma(a), x \in \gamma(u)$, $\mathsf{c}'xy \in \gamma(u * [a])$. Let $\mathsf{q}_0, \mathsf{q}_1$ be such that for all $x \in \gamma(a)$, $\mathsf{q}_0 x \in \gamma(\mathsf{p}_0 a)$, $\mathsf{q}_1 x \in \gamma(\mathsf{p}_1 a)$. By the recursion theorem, let $e \in B$ be such that

$$exv \precsim \text{ if } \mathsf{d}\mathsf{q}_0(re(\mathsf{c}xv)) \text{ then } \mathsf{q}_1(re(\mathsf{c}xv)) \tag{14}$$

$$\text{else } ex(\mathsf{c}'v(r_F(r(\mathsf{q}_1(re(\mathsf{c}xv)))))). \tag{15}$$

Then one can verify that for $v \in \gamma([])$, we have for all $e \in \gamma(a)$, $x \in \gamma(a')$ such that $a \cdot^f a' \downarrow$:

$$exv \in \gamma(a \cdot^f a').$$

Essential is that if $a \in I_1^{A[F]}(f)$, then for $b \in \gamma(a)$, $rb \in I_1^\gamma(f)$.

So by the above, $\langle xy \rangle xyv$ realizes γ as an applicative morphism $\gamma_F : A[F] \to B$, and the fact that $\gamma = \gamma_F \circ \iota_F$ is obvious.

Uniqueness is easy to check. □

Remark 3.4. The construction of $A[F]$ can be thought of as a construction in stages: we say that $a \cdot^F b\downarrow$ *at stage* α if $a \cdot^{f_\alpha} b\downarrow$. In proofs of statements about $A[F]$, we will often use induction on the stage at which a term denotes. The infinitary nature of computation relative in a functional, as remarked by Kleene in [8], is reflected by this construction in (transfinitely many) stages.

Example 3.5. We again take a look at the functional E from example 3.2. For any pca A, E is an effective operation in $A[E]$. As a consequence, the computable predicates in $A[E]$ are closed under existential quantification: suppose $p \in A[E]$ is a total function in 2 variables, taking values in $\{\mathsf{T}, \mathsf{F}\}$. Then:

$$Q(n) := (\exists m) pnm$$

is clearly also a recursive predicate in $A[E]$, so it has some index $q \in A[E]$.

For $A = \mathcal{K}_1$, this produces a familiar (see [8]) result. Namely, it follows (by taking complements) that every arithmetical subset of $\mathbb{N}$ is computable in $\mathcal{K}_1[E]$. We will see below that the total functions in $\mathcal{K}_1[E]$ are in fact precisely the hyperarithmetical functions.

Remark 3.6. One may want to define, for functionals $F, G : A^A \to A$, a pca $A[F, G]$ in which F, G are both effective operations. It will not work to define this as $A[F][G]$. However, if one codes any tuple of functionals into one functional H (e.g. $H(g) = \mathsf{p}F(g)G(g)$), then $A[H]$ will satisfy the required properties and the analogue of theorem 2.7 for such a tuple holds.

4 Effective Operations and Kleene computability

In [8], Kleene developed a recursion theory for functionals, which included a notion of computability relative to a functional. A short overview of this theory, as well as comparisons to other notions of higher-type computation can be found in [10].

In this section we will show that a total function $f : \mathbb{N} \to \mathbb{N}$ is computable relative to a functional $F : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ (in Kleene's sense) if and only if it has an index in $\mathcal{K}_1[F]$. This result will be almost immediate, since (as we will see) Kleene's indexed set of functions computable in F is basically an explicit model of $\mathcal{K}_1[F]$. Our construction has the advantage of being independent of any explicit model of $\mathcal{K}_1$ and avoids a lot of technicalities. Moreover, there is the immediate generalization to arbitrary pcas and the connection to realizability.

In his paper, Kleene defined an $\mathbb{N}$-indexed set of functions that may take tuples of arguments of all finite types. The definition consists of nine inductive rules, named S1-S9. A function with index e is denoted by $\{e\}$. A function $g : \mathbb{N} \to \mathbb{N}$ is then computable *relative to* the functional $F : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ if there is an index $e \in \mathbb{N}$ such that for all $n \in \mathbb{N}$:

$$\{e\}(n, F) = g(n).$$

Instead of defining Kleene's S1-S9 in full generality, we will just state eight derived rules that define the indexed set of functions computable relative to some functional F according to Kleene. The rule S7 is missing since that applies specifically to type 1 arguments, and we only consider a specific type 2 oracle.

Definition 4.1. Let $F : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ be a functional. Also fix some primitive recursive coding of tuples $\langle - \rangle : \mathbb{N}^{<\omega} \to \mathbb{N}$. We define an indexed set of partial functions $\mathbb{N}^{<\omega} \to \mathbb{N}$ by the following rules: for all $n, m \in \mathbb{N}$:

S1 Successor function: $\{\langle 1, 1\rangle\}(n) = n + 1$

S2 Constant function: $\{\langle 2, 1, m\rangle\}(n) = m$

S3 Projection: for all $n_1, \ldots, n_m \in \mathbb{N}$, $\{\langle 3, m\rangle\}(n_1, \ldots, n_m) = n_1$

S4 Composition: For $g, h \in \mathbb{N}$, all $k, n_1, \ldots, n_k \in \mathbb{N}$:

$$\{\langle 4, k, g, h\rangle\}(n_1, \ldots, n_k) \simeq \{g\}(\{h\}(n_1, \ldots, n_k), n_1, \ldots, n_k).$$

S5 Primitive recursion: For any $g, h \in \mathbb{N}$, if $m = \langle 5, 2, g, h\rangle$ then for all k:

$$\{m\}(0, n) \simeq \{g\}(n) \tag{16}$$

$$\{m\}(k + 1, n) \simeq \{h\}(k, \{m\}(k, n), n) \tag{17}$$

S6 Permutation of arguments: For any $g \in \mathbb{N}, 1 \leq k < r$ and $n_1, \ldots, n_r \in \mathbb{N}$:

$$\{\langle 6, r, k, g\rangle\}(n_1, \ldots, n_r) \simeq \{g\}(n_{k+1}, n_1, \ldots, n_k, n_{k+2}, \ldots, n_r).$$

S8 Application of F: For every $h, e \in \mathbb{N}$:

$$\{\langle 8, 1, h\rangle\}(n) \simeq F(\lambda k.\{h\}(k, n))$$

S9 Index vocation: for all $m, k, l, n_1, \ldots, n_{k+l} \in \mathbb{N}$

$$\{\langle 9, k, l\rangle\}(m, n_1, \ldots, n_{k+l}) \simeq \{m\}(n_1, \ldots, n_k).$$

Note. Here, $a \simeq b$ has the same meaning as before: the right-hand side is defined if and only if the left-hand side is defined, and when this is the case both sides are equal.

In what follows, we assume that we have fixed some functional $F : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$, and defined the partial map $\{-\}(-) : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ according to the above.

Theorem 4.2. *The partial map* $\{-\}(-) : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *defines a partial combinatory algebra* $\mathcal{K}_1^F$.

Proof. Tedious exercise using the clauses in definition 4.1. □

The following can be viewed as the type 2 analogue of corollary 2.9(iii).

Theorem 4.3. *There is an isomorphism of pcas $\mathcal{K}_1[F] \cong \mathcal{K}_1^F$.*

Proof. By Church's thesis, the identity relation on $\mathbb{N}$ is an applicative morphism $\gamma : \mathcal{K}_1 \to \mathcal{K}_1^F$. It is easy to see that $F \in E_\gamma$. By theorem 3.3, the identity relation on $\mathbb{N}$ is an applicative morphism $\gamma_F : \mathcal{K}_1[F] \to \mathcal{K}_1^F$.

It is not hard to show that we can define the application in $\mathcal{K}_1^F$ computably in $\mathcal{K}_1[F]$ using the clauses 4.1 and the recursion theorem when needed. So the identity relation on $\mathbb{N}$ is also an applicative morphism $\mathcal{K}_1^F \to \mathcal{K}_1[F]$. Therefore $\mathcal{K}_1[F] \cong \mathcal{K}_1^F$. □

Corollary 4.4. *Let E be as in example 3.2 for $A = \mathcal{K}_1$. The total functions in $\mathcal{K}_1[E]$ are precisely the hyperarithmetical functions.*

Proof. This follows from the same fact for $\mathcal{K}_1^E$ and theorem 4.3, which is theorem XLVIII in [8]. □

The above corollary is what we can deduce from Kleene's original paper on recursive functionals. However, we can do a little better using the results of Hinman's approach in [4]. There, the domains of the $\mathbb{N}-$indexed partial functions $\mathbb{N} \to \mathbb{N}$ "computable in E" are precisely the Π^1_1-sets. Here we can prove something similar. In the context of pcas and applicative morphisms, domains of partial recursive functions are not particularly well-behaved (we have to impose more conditions on morphisms to preserve this structure, see for example the discussion of dominances in [9]). Theorem 5.8 also exemplifies this fact. We therefore avoid domains, and show the following instead:

Proposition 4.4.1. The application in $\mathcal{K}_1[E]$ is Π^1_1-complete.

Proof. We show here that the application function is Π^1_1, since that does not follow immediately from Hinman's results. Let $r \in \mathbb{N}$ be the uniform index for f in $\mathcal{K}_1[f]$, for any f. Recall that $\mathcal{K}_1[E] = \mathcal{K}_1[g]$, where we defined g in stages. I'll show that g is Π^1_1. Let e be an index such that whenever ϕ is the characteristic function of the graph of a partial function $f : A \to A$, then for all $b, x \in \mathcal{K}_1$:

$$eb \cdot^\phi x \simeq b \cdot^f x.$$

Convince yourself that such e exists; since $\mathbb{N}$ is enumerable, we can just "wait" for values of f to appear in the graph.

Define the following predicate on $\mathbb{N} \times \mathbb{N}^\mathbb{N}$:

$$\Phi([b,i], X) = (\forall s \exists t \forall y) : \big(eb \cdot^X s\downarrow \wedge (eb \cdot^X t = 0 \vee i = 1) \wedge (eb \cdot^X y \neq 0 \vee i = 0).\big) \tag{18}$$

It is not hard to see that Φ is Π^1_1 (in fact it is arithmetical) (to work out the details, one needs to use Kleene's $T-$predicate and existential quantification over dialogues). Let G be the characteristic function of the graph of g. Now I claim:

$$G([b,i]) \iff (\forall X)(\exists p, j)(\Phi([p,j], X) \Rightarrow X([p,j])) \Rightarrow X([b,i]). \tag{19}$$

Proof of claim. First, observe:

$$\begin{aligned}
\Phi([b,i],G) &\Leftrightarrow (\forall s)(\exists t \forall y) eb \cdot^{G} s \downarrow \wedge (eb \cdot^{G} t = 0 \vee i = 1) \wedge (eb \cdot^{G} y \neq 0 \vee i = 0) && (20)\\
&\Leftrightarrow (\forall s)(\exists t \forall y) b \cdot^{g} s \downarrow \wedge (b \cdot^{g} t = 0 \vee i = 1) \wedge (b \cdot^{g} y \neq 0 \vee i = 0) && (21)\\
&\Leftrightarrow G([b,i]). && (22)
\end{aligned}$$

This proves the "$\Leftarrow$" direction of (19) (we substitute G for X).

Now assume $G([b,i])$, that is:

$$(\forall s)(\exists t \forall y) b \cdot^{g} s \downarrow \wedge (b \cdot^{g} t = 0 \vee i = 1) \wedge (b \cdot^{g} y \neq 0 \vee i = 0).$$

Suppose we have X such that

$$(\forall p, j) : \Phi([p,j], X) \Rightarrow X([p,j]).$$

We need to prove that $X([b,i])$. I'll prove this by showing that

$$(\forall p, j) : G([p,j]) \Rightarrow X([p,j]). \qquad (23)$$

This is done by induction on stages (see remark 3.4). Recall that g is constructed as $g = \bigcup_\alpha g_\alpha$. Denote the corresponding graphs by G_α. Then for $\alpha = 0$,

$$(\forall p, j) : G([p,j]) \Rightarrow X([p,j])$$

clearly holds, since $g_0 = \emptyset$. If it holds up to α, then:

$$\begin{aligned}
G_{\alpha+1}[p,j] &\Rightarrow (\forall s)(\exists t \forall y) p \cdot^{g_\alpha} s \downarrow \wedge (p \cdot^{g_\alpha} t = 0 \vee j = 1) \wedge (p \cdot^{g_\alpha} y \neq 0 \vee j = 0) && (24)\\
&\Rightarrow (\forall s)(\exists t \forall y) ep \cdot^{X} s \downarrow \wedge (ep \cdot^{X} t = 0 \vee j = 1) \wedge (ep \cdot^{X} y \neq 0 \vee j = 0) && (25)\\
&\Rightarrow \Phi([p,j], X) && (26)\\
&\Rightarrow X([p,j]). && (27)
\end{aligned}$$

For limit ordinals λ, $G_\lambda[p,j] \iff G_\alpha[p,j]$ for some $\alpha < \lambda$ so that follows trivially.

Hence we have shown (23), from which $X[b,i]$ follows by assumption. □

Since the right-hand side of (19) is Π^1_1, we are done.

To see that the application is Π^1_1 complete, see [4], section VI.1. There one can find a proof in which a well-known Π^1_1-complete set is reduced to a *semi-recursive* set computable from E in Hinman's sense (i.e. a domain of a partial recursive function). This proof is easily adopted to a proof of Π^1_1-completeness for our case. □

5 Local Operators

In [15], a notion of realizability is studied that uses a specific local operator J on the Effective Topos. We will show here that some of the results in that paper can be derived easily using the functional E from example 3.2 (for $A = \mathcal{K}_1$) and theorem 3.3.

We first give a definition of local operators in realizability toposes on a pca A, that we assume fixed from now on. The topos-theoretic origin is omitted here, for more about that we refer to [14] or [12]. We introduce the following notation: For $P, Q \subseteq A$ subsets, let

$$P \to Q = \{a \in A \,|\, (\forall b \in P) ab\downarrow \text{ and } ab \in Q\}$$

Definition 5.1. A *local operator* is a function $J : \mathcal{P}(A) \to \mathcal{P}(A)$ such that there are $e_1, e_2, e_3 \in \mathbb{N}$ that satisfy:

1. $e_1 \in \bigcap_{P \subseteq A} P \to JP$
2. $e_2 \in \bigcap_{P \subseteq A} JJP \to JP$
3. $e_3 \in \bigcap_{P,Q \subseteq A} (P \to Q) \to (JP \to JQ)$

We define a pre-order on local operators as follows: we say $J \leq J'$ whenever there is $e \in A$ such that:

$$e \in \bigcap_{P \subseteq A} JP \to J'P.$$

We write $J \cong J'$ if $J \leq J'$ and $J' \leq J$. There is a least local operator $J_\perp$, given by the identity: $J_\perp(P) = P$. There is also a biggest local operator given by $J_\top(P) = A$.

The lattice of local operators (up to isomorphism) corresponds to the lattice of subtoposes of the realizability topos on A, denoted $\mathrm{RT}(A)$. Some of these subtoposes are realizability toposes $\mathrm{RT}(B)$ themselves, in that case we have found a *geometric inclusion* $\mathrm{RT}(B) \to \mathrm{RT}(A)$.

Definition 5.2 ([5])**.** An applicative morphism $\gamma : A \to B$ is called *computationally dense* if there exists $m \in B$ such that

$$(\forall b \in B)(\exists a \in A)(\forall a') b\gamma(a') \downarrow \Rightarrow m\gamma(aa') \subseteq b\gamma(a').$$

Geometric inclusions $\mathrm{RT}(B) \to \mathrm{RT}(A)$ correspond precisely to computationally dense applicative morphisms $\gamma : A \to B$ that satisfy the additional condition

$$\text{(in) } (\exists e)(\forall b)(\exists a)(eb \in \gamma(a) \text{ and } m\gamma(a) = \{b\})$$

where m is as in definition 5.2 (this is a consequence of results in [5, 7]). In that case the corresponding local operator in $\mathrm{RT}(A)$ is given by

$$J(P) = \{a \in A \,|\, m\gamma(a) \subseteq \bigcup_{a' \in P} \gamma(a')\}.$$

The following corresponds to proposition 2.2 in [13].

Proposition 5.2.1. For $f : A \to A$ a partial function, the identity relation on A is a computationally dense applicative morphism $\iota_f : A \to A[f]$ that satisfies the condition (in). Therefore, there is a canonical geometric inclusion $\mathrm{RT}(A[f]) \to \mathrm{RT}(A)$.

By our construction, we obtain for every (partial) functional $F : A^A \to A$ a subtopos $\mathrm{RT}(A[F]) \to \mathrm{RT}(A)$ corresponding to the local operator

$$J_F(P) = \{a \in A \,|\, m \cdot^F a \in P\}.$$

One can check that this indeed defines a local operator according to definition 5.1.

5.1 J-Assemblies

In [6], Martin Hyland showed that for every function $f : \mathbb{N} \to \mathbb{N}$, there is a least local operator that forces f to be computable in the Effective Topos. This yields an embedding of the Turing degrees into the lattice of local operators. In [11], Wesley Phoa showed that the corresponding subtopos is equivalent to $\mathrm{RT}(\mathcal{K}_1^f)$, the realizability topos on the pca of recursive application with oracle f. This result can be generalized to arbitrary pcas and partial functions: for (partial) functions $f : A \to A$, the least subtopos of $\mathrm{RT}(A)$ in which f is realizable is equivalent to $\mathrm{RT}(A[f])$. For a proof, see [1].

We will now carry out a similar statement for effective operations. To make things easier, we will work on a category of assemblies. The consequences for realizability toposes are immediate for anyone familiar with them. In the following we assume that we have fixed some pca A.

Definition 5.3. Let J be a local operator for A. A J-*assembly* is a pair (X, E) where $E : X \to A$ is a total relation.

For $(X, E), (Y, F)$ J-assemblies, a *morphism* of $J-$assemblies is a function $f : X \to Y$ together with a $t \in A$ such that

$$t \in \bigcap_{x \in X} E(x) \to JF(f(x)).$$

We say that t *tracks* f.

One can check that the composition $g \circ f$ of morphisms of $J-$assemblies $f : (X, E) \to (Y, F)$, $g : (Y, F) \to (Z, G)$ is again a morphism of J-assemblies, so that $J-$assemblies form a category $\mathrm{Ass}(A)_J$. The category $\mathrm{Ass}(A)_J$ has a lot of structure: it is regular, cartesian closed and has finite colimits. For (X, E), (Y, F) J-assemblies, the exponential $(Z, G) := (Y, F)^{X,E}$ is given by:

$$Z = \{f : X \to Y \,|\, f \text{ is a morphism } (X, E) \to (Y, F)\} \tag{28}$$

$$G(f) = \{t \in A \,|\, t \text{ tracks } f\}. \tag{29}$$

The assembly $\mathbf{A} = (A, \mathrm{id}_A)$ can be seen as a representation of the pca A in the category of J-assemblies. The object $\mathbf{A}^{\mathbf{A}}$ consists of the morphisms $\mathbf{A} \to \mathbf{A}$

in $\mathrm{Ass}(A)_J$. For $J = J_\perp$, these are precisely the computable functions in A. Also, one can check that the arrows

$$\mathbf{A}^{\mathbf{A}} \to \mathbf{A}$$

in $\mathrm{Ass}(A)_{J_\perp}$ are precisely the effective operations in A. We have the following theorem:

Theorem 5.4. *Let $F : A^A \to A$ be a total functional, and assume that F defines a morphism*

$$\mathbf{A}^{\mathbf{A}} \to \mathbf{A}$$

in $\mathit{Ass}(A)_J$. Then $J_F \leq J$.

Proof. We can assume that J preserves inclusions, otherwise we let

$$J'P = \bigcup_{Q \subseteq P} JP$$

and show that $J' \cong J$, which is an easy exercise using definition 5.1.

By computational density, there exists $t \in A$ such that for all $v, x \in A$:

$$m \cdot^F (tvx) \lesssim v \cdot^F x.$$

Since m is independent of F, we also know that if the right side halts at some stage, the left hand side also halts at that stage.

We pick e_1, e_2, e_3 as in definition 5.1.

Suppose that $F : \mathbf{A}^{\mathbf{A}} \to \mathbf{A}$ is tracked by r as morphism of J-assemblies.

Choose $\mathsf{m} \in A$ (using the recursion theorem) so that:

$$\mathsf{m}xy \lesssim \mathsf{if}\ \mathsf{p}_0(m([x] * y)) \tag{30}$$
$$\mathsf{then}\ e_1(\mathsf{p}_1(m([x] * y))) \tag{31}$$
$$\mathsf{else}\ e_2\,(e_3(\langle z\rangle \mathsf{m}x(y * [z]))r(\langle w\rangle(\mathsf{m}(tvw)[]))) \tag{32}$$
$$\text{where } v = p_1(m([x] * y)). \tag{33}$$

Here [] is the empty sequence. We will prove by induction on the stage at which $m \cdot^F x \downarrow$ that for all x,

$$\mathsf{m}x[] \in J\{m \cdot^F x\}. \tag{34}$$

Stage 0: in that case $m([x]) = \mathsf{pT}c$ for some c. Then

$$\mathsf{m}x[] = e_1(\mathsf{p}_1(m([x] * []))) \in J\{c\}. \tag{35}$$

Stage $\lambda > 0$: Let $u = [u_0, \ldots, u_n]$ be the halting dialogue between m and x. It is easy to see that

$$\mathsf{m}x[u_0, \ldots, u_n] \in J\{m \cdot^F x\}.$$

Now suppose that $0 < i \leq n + 1$ is such that

$$\mathsf{m}([x] * u^{<i+1}] \in J\{m \cdot^F x\}. \tag{36}$$

We know that:

$$m([x]*u^{<i}) = \mathsf{pF}v$$

where v is such that at some stage $\alpha < \lambda$, for each w, $v \cdot^F w \downarrow$. So for each w, also $m \cdot^F (tvw) \downarrow$ at stage α. Let f be the function $w \mapsto v \cdot^F w$. By induction hypothesis,

$$\mathsf{m}(tvw)[] \in J\{v \cdot^F w\}$$

for each w, so $\langle w\rangle \mathsf{m}(tvw)[]$ tracks f as morphism of $J-$assemblies. So

$$r(\langle w\rangle(\mathsf{m}(tvw)l)) \in J\{F(f)\}$$

and $F(f) = u_i$. By (36), we have:

$$\langle z\rangle \mathsf{m}([x]*u^{<i}*[z]) \in \{F(f)\} \to J\{m \cdot^F x\}$$

and therefore

$$e_2\left(e_3(\langle z\rangle \mathsf{m}x(u^{<i}*[z]))r(\langle w\rangle(\mathsf{m}(tvw)[]))\right) \in J\{m \cdot^F x\}$$

hence it follows that

$$\mathsf{m}([x]*u^{<i}) \in J\{m \cdot^F x\}. \tag{37}$$

By reverse induction, it follows that

$$\mathsf{m}x[] \in J\{m \cdot^F x\},$$

so this finishes the induction step, so we have (34).

Therefore

$$\langle x\rangle \mathsf{m}x[] \in \bigcap_{P\subseteq A} J_F P \to JP$$

and we are done. □

5.2 Pitts' local operator

In this section we will take a look at the operator J that is studied in [15]. This operator was introduced by Andrew Pitts in his thesis (example 5.8, [12]). Denote by $\nabla(\mathbb{N})$ the assembly

$$(\mathbb{N}, T) \text{ where } T(n) = \mathbb{N} \text{ for all } n.$$

We can define the following subobject of $\nabla(\mathbb{N})$:

$$(\mathbb{N}, R) \rightarrowtail \nabla(\mathbb{N}) \text{ where } R(n) = \{m \in \mathbb{N} \mid m \geq n\}.$$

Pitts' local operator J is defined as the least local operator in $\mathrm{RT}(\mathcal{K}_1)$ that forces the above arrow to be an isomorphism of $J-$assemblies. Explicitly, it is given by:

$$J(P) = \bigcap\{Q \subseteq \mathbb{N} \mid \{0\} \wedge P \subseteq Q \text{ and } \{1\} \wedge (\bigcup_{n\in\mathbb{N}}(R(n) \to Q)) \subseteq Q\}$$

where for $R, S \subseteq \mathbb{N}$:

$$R \wedge S = \{\mathsf{p}rs \,|\, r \in R, s \in S\}.$$

Write $\mathbf{N}$ for the assembly $\mathbf{A}$ for $A = \mathcal{K}_1$, so

$$(\mathbb{N}, N) \text{ where } N(n) = \{n\}.$$

Recall the functional E from example 3.2. For $A = \mathcal{K}_1$, we rather define it as

$$E(f) = \begin{cases} 0 & \text{if } (\exists n) f(n) \\ 1 & \text{otherwise.} \end{cases}$$

We have the following proposition:

Proposition 5.4.1. There is an index $r_E \in \mathbb{N}$ that tracks E as morphism of J-assemblies $\mathbf{N}^{\mathbf{N}} \to \mathbf{N}$.

Proof. Pick e_3 as in definition 5.1.

By corollary 1.6 of [15], there is a partial recursive function G such that for all

$$x_0 \in J\{a_0\}, \ldots, x_{n-1} \in J\{a_{n-1}\},$$

we have

$$G(\langle x_0, \ldots, x_{n-1} \rangle) \in J\{0\} \text{ if for some } i < n, a_i = 0 \tag{38}$$

$$G(\langle x_0, \ldots, x_{n-1} \rangle) \in J\{1\} \text{ otherwise} \tag{39}$$

Now let t_E be an index in $\mathcal{K}_1$ defined by:

$$t_E en \simeq G(\langle e(0), \ldots, e(n) \rangle)$$

Then whenever e realizes a total function $f : \mathbf{N} \to \mathbf{N}$, we $(\exists n) f(n) = 0$ if and only if $(\exists n) en \in J\{0\}$, which holds if and only if there is n such that

$$t_E e \in \{m \,|\, m \geq n\} \to J\{0\}$$

since otherwise

$$t_E e \in \{m \,|\, m \geq 0\} \to J\{1\}.$$

By definition of J, we have in the first case that $t_E e \in J\{0\}$, and in the latter case $t_E e \in J\{1\}$. Now t_E tracks E as morphism of $J-$assemblies $\mathbf{N}^{\mathbf{N}} \to \mathbf{N}$. □

The following corollary corresponds to theorem 2.2 in [15].

Corollary 5.5. *The morphisms of J-assemblies $\mathbf{N} \to \mathbf{N}$ are precisely the hyperarithmetical functions.*

Proof. By theorem 5.4, $J_E \leq J$. Therefore it is easily seen that every morphism of J_E assemblies $\mathbf{N} \to \mathbf{N}$ is a morphism of $J-$assemblies. Now it follows that every hyperarithmetical function is a morphism of J-assemblies by corollary 4.4.

For the converse, observe that every morphism $f : \mathbf{N} \to \mathbf{N}$ is Π^1_1 since:

$$f(n) = m \iff (\forall B)\{\mathsf{p}0m\} \subseteq B \wedge \{1\} \wedge (\{e \,|\, (\exists n)(\forall m \geq n)em \in B\}) \subseteq B \to e(\mathsf{a}n) \in B$$

where e is a realizer for f as a morphism $\mathbf{N} \to \mathbf{N}$. Notice that the above expression is Π^1_1. Since f is total, it is also Σ^1_1.

Therefore the morphisms of J-assemblies $\mathbf{N} \to \mathbf{N}$ are precisely the hyperarithmetical functions. □

5.3 J-representable functions

In [15], the morphisms of J-assemblies $\mathbf{N} \to \mathbf{N}$ are called $J-$*representable.* Since for every $n, m \in \mathbb{N}$

$$n \neq m \iff J\{n\} \cap J\{m\} = \emptyset, \tag{40}$$

the following is well defined:

Definition 5.6. A partial function $F : \mathbb{N} \to \mathbb{N}$ is $J-$*representable* if there exists $e \in \mathbb{N}$ such that for all $m \in \mathbb{N}$:

$$\phi_e(n) \in J\{m\} \iff F(n) = m.$$

Here ϕ_e denotes the $e-$th partial computable function from ordinary recursion theory (or the application in $\mathcal{K}_1$).

By defining $\psi_e = F$ for such e, we obtain an indexed set of partial functions and a new application

$$e \cdot n \mapsto \psi_e(n)$$

on $\mathbb{N}$.

A natural question is whether the application $e \cdot n \simeq \psi_e(n)$ is combinatory complete in the sense of definition 2.3. This turns out to be the case:

Proposition 5.6.1. The application $e \cdot n \simeq \psi_e(n)$ defines a pca J_1 on $\mathbb{N}$.

Proof. This is an easy exercise on pcas and definition 5.1. □

The above proposition would be particularly hard to prove (if not false) if one uses the strict version of the definition of a pca. In the next section we will see that we do not have to worry about this.

5.4 Strict Pcas

As announced in the introduction, several authors use a stronger notion of combinatory completess, that we shall call *strictly combinatory complete*:

Definition 5.7. Let A be a partial applicative structure. Then A is called a *strict pca* when there are $\mathsf{k}, \mathsf{s} \in A$ such that for all $a, b, c \in A$:

1. $\mathsf{s}ab\downarrow$
2. $\mathsf{k}ab = a$
3. $\mathsf{s}abc \simeq ac(bc)$.

We also call the applicative structure on A *strictly combinatory complete.*

This definition might be preferable in some cases, for example when one would like that the set of domains of partial computable functions in a pca A form a *dominance* in $\mathrm{RT}(A)$(see [9]). However, as mentioned before, these domains are not part of the structure of a pca in the category PCA.

The following theorem supports this: in the context of applicative morphisms, the notions of a pca and a strict pca are essentially the same.

Theorem 5.8. *Every pca is isomorphic to a strict pca.*

Proof. Let A be a pca. If A is trivial, then the statement is obvious, so we assume A non-trivial.

The idea is to define an explicit pca structure A' on the set A, such $a \mapsto \{a\}$ is an applicative morphism $A' \to A$. In this explicit structure, we also make sure that we can compute the application in A.

To define the new applicative structure on A, we use tuples and Curry numerals from A to ensure that what we define is actually computable in A. We denote the new application by $\cdot : A \times A \to A$, and the resulting partial applicative structure by A'. The definition is by induction on the following clauses:

1. Constant function: for all $a, b, \in A$:

$$[\overline{1}] \cdot a = [\overline{1}, a] \tag{41}$$

$$[\overline{1}, a] \cdot b = a \tag{42}$$

2. S-combinator: For $a, b, c \in A$:

$$[\overline{2}] \cdot [a, b, c] \simeq a \cdot c \cdot (b \cdot c)$$

3. Currying of parameters: for all $e, a, b, c \in A$:

$$[\overline{3}, e] \cdot a = [\overline{3}, e, a] \tag{43}$$

$$[\overline{3}, e, a] \cdot b = [\overline{3}, e, a, b] \tag{44}$$

$$[\overline{3}, e, a, b] \cdot c \simeq e \cdot [a, b, c] \tag{45}$$

4. Application in A:

$$[\overline{4}] \cdot a = [\overline{4}, a] \tag{46}$$

$$[\overline{4}, a] \cdot b \simeq ab \tag{47}$$

Since A is non-trivial, the above is well-defined. Strict combinatory completeness of A is satisfied by taking $\mathsf{k}' = [\overline{1}]$, $\mathsf{s}' = [\overline{3}, [\overline{2}]]$. It is easy to see that $a \mapsto \{a\}$ can be realized as applicative morphism $A' \to A$, and as applicative morphism $A \to A'$ it is realized by $[\overline{4}]$. □

Remark 5.9. Note that the proof above is entirely constructive; to define the application it only uses natural induction and explicit constructions with the combinators k and s.

Another observation is that if we would define A' using only the clauses 1-3, we obtain a strict pca A', together with an applicative morphism $A' \to A$ given by $a \mapsto \{a\}$, that has "forgotten" a lot of the structure of A. We can then adjoin certain partial functions in A to A' to get some of them back (such as the application, so that we obtain A again). In the case of a decidable pca, for instance, we would lose decidability in this way. So, not only can we adjoin functions to a pca as an oracle, we are also able to forget (all but some) functions in a pca.

Finally, we have the following theorem:

Theorem 5.10. *The pca J_1 is isomorphic to $\mathcal{K}_1[E]$.*

Proof. Let $\gamma : \mathcal{K}_1 \to J_1$ be the applicative morphism $n \mapsto \{n\}$ (it is easily verified that this is a decidable applicative morphism).

Now $E \in E_\gamma$ by proposition 5.4.1. Therefore, by theorem 3.3, γ factors through $\mathcal{K}_1[E]$, hence we have a decidable applicative morphism $\gamma_E : \mathcal{K}_1[E] \to J_1$, given by the identiy on $\mathbb{N}$.

This morphism also goes the other way: this follows from the fact that the application in $\mathcal{K}_1[E]$ is Π^1_1-complete (proposition 4.4.1), and the fact that the application in J_1 is Π^1_1. □

6 Higher types

One may wonder whether theorem 4.3 also holds for computability in higher type objects (so, type 3 and higher). The answer to that question is positive: for functionals $x_1, \ldots, x_n$ of certain types, one can define in a similar fashion a pca

$$\mathcal{K}_1[x_1, \ldots, x_n]$$

that is again isomorphic to the pca of recursive application relative relative to $x_1, \ldots, x_n$ in Kleene's sense. The idea is to define $\mathcal{K}_1[x_1, \ldots, x_n] = \mathcal{K}_1[f]$, where f is defined in stages, simultaneously for $x_1, \ldots, x_n$, so that the resulting applicative structure is closed under application of any of the functionals.

However, the connection with effective operations does not generalize in this way. Furthermore, it is not known (at least to me) what an analogue of theorem 5.4 would be for, say, a type 3 functional. This might be interesting to find out.

References

[1] Eric Faber and Jaap van Oosten. More on Geometric Morphisms between Realizability Toposes. 2014. Submitted.

[2] Jonas Frey. A characterization of realizability toposes. 2014. Submitted to Journal of Pure and Applied Algebra.

[3] R. O. Gandy and J. M. E. Hyland. Computable and recursively countable functions of higher type. In *Logic Colloquium 76 (Oxford, 1976)*, pages 407–438. Studies in Logic and Found. Math., Vol. 87. North-Holland, Amsterdam, 1977.

[4] Peter G. Hinman. *Recursion-theoretic hierarchies*. Springer-Verlag, Berlin, 1978. Perspectives in Mathematical Logic.

[5] Pieter Hofstra and Jaap van Oosten. Ordered partial combinatory algebras. *Math. Proc. Cambridge Philos. Soc.*, 134(3):445–463, 2003.

[6] J. M. E. Hyland. The effective topos. In *The L.E.J. Brouwer Centenary Symposium (Noordwijkerhout, 1981)*, volume 110 of *Stud. Logic Foundations Math.*, pages 165–216. North-Holland, Amsterdam, 1982.

[7] Peter Johnstone. Geometric morphisms of realizability toposes. *Theory Appl. Categ.*, 28:No. 9, 241–249, 2013.

[8] S. C. Kleene. Recursive functionals and quantifiers of finite types. I. *Trans. Amer. Math. Soc.*, 91:1–52, 1959.

[9] John R. Longley. *Realizability Toposes and Language Semantics*. PhD thesis, University of Edinburgh, 1995.

[10] John R. Longley. Notions of computability at higher types. I. In *Logic Colloquium 2000*, volume 19 of *Lect. Notes Log.*, pages 32–142. Assoc. Symbol. Logic, Urbana, IL, 2005.

[11] Wesley Phoa. Relative computability in the effective topos. *Math. Proc. Cambridge Philos. Soc.*, 106(3):419–422, 1989.

[12] A.M. Pitts. *The Theory of Triposes*. PhD thesis, Cambridge University, 1981.

[13] Jaap van Oosten. A general form of relative recursion. *Notre Dame J. Formal Logic*, 47(3):311–318 (electronic), 2006.

[14] Jaap van Oosten. *Realizability: an introduction to its categorical side*, volume 152 of *Studies in Logic and the Foundations of Mathematics*. Elsevier B. V., Amsterdam, 2008.

[15] Jaap van Oosten. Realizability with a local operator of A.M. Pitts. *Theoret. Comput. Sci.*, 546:237–243, 2014.