EMBEDDINGS BETWEEN PARTIAL COMBINATORY ALGEBRAS

ANTON GOLOV AND SEBASTIAAN A. TERWIJN

Abstract. Partial combinatory algebras are algebraic structures that serve as generalized models of computation. In this paper, we study embeddings of pcas. In particular, we systematize the embeddings between relativizations of Kleene's models, of van Oosten's sequential computation model, and of Scott's graph model, showing that an embedding between two relativized models exists if and only if there exists a particular reduction between the oracles. We obtain a similar result for the lambda calculus, showing in particular that it cannot be embedded in Kleene's first model.

1. Introduction

Partial combinatory algebras were introduced by Feferman [7] to study generalized notions of computability. Feferman's motivation was his study of predicative theories (explicit mathematics), but the interest in pcas extends to well beyond that. For the role of pcas in constructive mathematics, the reader may consult the monographs by Beeson [3] and Troelstra and van Dalen [25]. In particular, pcas have been used as a basis for models of constructive set theory, see McCarty [14], Rathjen [20], and Frittaion and Rathjen [9]. They also play an important role in the theory of realizability, as documented in van Oosten [18].

Combinatory algebras (i.e. the total versions of pcas) were studied long before Feferman, and even predate the lambda calculus, to which they are closely connected. For the connections between these subjects we refer the reader to Barendregt [2]. Even though pcas were not defined explicitly in that context, the notion of partiality was studied there extensively as well. Given the connection with lambda calculus, there has been an obvious interest in pcas from this direction. For example, the topic of completions of pcas was studied by Klop and Bethke and others, cf. [4] and [5].

In this paper we study embeddings between pcas. Such embeddings have been studied before, with various notions of embedding (explained below). Though we also consider uncountable pcas, our results concentrate on countable pcas, some of which are derived from the uncountable examples. Well-studied examples of pcas are Kleene's first and second models $\mathcal{K}_1$ and $\mathcal{K}_2$, van Oosten's sequential computation model $\mathcal{B}$ [19], and Scott's graph model $\mathcal{G}$ [21], as well as the relativized versions of these. We relate the existence

Date: November 25, 2022.

2010 *Mathematics Subject Classification.* 03D28, 03B40, 03D80.

Key words and phrases. partial combinatory algebra, embeddings, Turing degrees, enumeration degrees.

of embeddings between relativizations of these pcas with the existence of Turing and enumeration reductions between the oracles.

A partial combinatory algebra (pca) $\mathcal{A}$ is a set A with a partial binary operation $\cdot$ satisfying a particular completeness condition. Every pca has elements representing the combinators s and k (cf. Definition 2.2 below), and the existence of such elements is sufficient for $\mathcal{A}$ to be a pca.

Given pcas $\mathcal{A} = (A, \cdot_A)$ and $\mathcal{B} = (B, \cdot_B)$, a natural notion of a pca morphism $\mathcal{A} \to \mathcal{B}$ is a function $f : A \to B$ that preserves the application when it is defined. An *embedding* $\mathcal{A} \hookrightarrow \mathcal{B}$ is such a function that is also injective.

This notion of embedding has previously been studied by Bethke [4], who showed that every pca can be embedded in a graph model, and by Shafer and Terwijn [22], who showed a number of embeddings between relativizations of Kleene's first and second models. In the context of completions, when application in $\mathcal{B}$ is required to be total, embeddings in this sense have been studied by Asperti and Ciabattoni [1]. Also in the context of completions, Bethke, Klop, and de Vrijer [5] studied a stronger notion of embedding, where the function $f : A \to B$ is additionally required to map s and k to the corresponding chosen elements in $\mathcal{B}$.

A more general notion of a map $\mathcal{A} \to \mathcal{B}$ was given by Longley [13] in the form of an applicative morphism. An applicative morphism γ is not required to preserve application; instead, there must exist a term in $\mathcal{B}$ that simulates the application on the image of γ. Additionally, applicative morphisms are not required to be single-valued. This notion of morphism is useful in the context of realizability, cf. van Oosten [18].

The following two results relating relativizations of $\mathcal{K}_1$ and $\mathcal{K}_2$ were proven in Shafer and Terwijn [22]:

Theorem 1.1 ([22, Proposition 8.2])**.** *Let $X, Y \subseteq \omega$. There is an embedding $\mathcal{K}_1^X \hookrightarrow \mathcal{K}_1^Y$ if and only if $X \leqslant_T Y$, and in this case the embedding is given by a computable function.*

Theorem 1.2 ([22, Theorem 8.5])**.** *For every $X \subseteq \omega$, there is an embedding $f : \mathcal{K}_1^X \hookrightarrow \mathcal{K}_2^X$. This embedding is X-computable, in the sense that there is an X-computable function g that maps n to a code of $f(n)$.*

We note here that the proofs of these theorems rely on the fact that the objects of these pcas can be coded by natural numbers. We extend this approach to obtain similar results about the other pcas introduced above. To this end, we generalize Ershov's notion of numbering [6] to allow for partiality, and define certain effectivity conditions on such partial numberings. We generalize these theorems in two ways. Taking $\mathcal{K}_1$ as an example, we get

(1) Given an arbitrary pca $\mathcal{A}$ equipped with a partial numbering $\gamma : \omega \to \mathcal{A}$, if the application in $\mathcal{A}$ can be represented using γ in an X-computable way, which moreover respects a chosen section of γ, then there is an embedding $\mathcal{A} \hookrightarrow \mathcal{K}_1^X$.

(2) Given an arbitrary pca $\mathcal{A}$ equipped with a partial numbering $\gamma : \omega \to \mathcal{A}$, if the application in $\mathcal{A}$ can be represented using γ in a Y-computable way and there is an embedding $\mathcal{K}_1^X \hookrightarrow \mathcal{A}$, then $X \leqslant_T Y$.

We focus in particular on the relativizations of Kleene's first and second models $\mathcal{K}_1$ and $\mathcal{K}_2$, of van Oosten's sequential computation model $\mathcal{B}$ ([19]), and of Scott's graph model $\mathcal{G}$ ([21]). For any X, the following sequence of embeddings holds:

$$\mathcal{K}_1^X \hookrightarrow \mathcal{K}_2^X \hookrightarrow \mathcal{B}^X \hookrightarrow \mathcal{G}^{X\oplus\overline{X}}.$$

The first is Theorem 1.2 while the second is given by the inclusion. The third is new (Theorem 7.3). Furthermore, we show that these embeddings are tight; that is, if for some Y, $\mathcal{K}_1^X \hookrightarrow \mathcal{K}_2^Y$ then $\mathcal{K}_2^X \hookrightarrow \mathcal{K}_2^Y$, and similarly for the other two (Theorems 6.7, 6.13, and 7.6).

The case where $X = \emptyset$ is of particular interest. Let $\mathcal{E}$ be the class of c.e. sets. These form a pca by Scott (see section 7 below). We have (Corollary 7.5) the sequence of embeddings

$$\mathcal{K}_1 \hookrightarrow \mathcal{K}_2^{\text{eff}} \hookrightarrow \mathcal{B}^{\text{eff}} \hookrightarrow \mathcal{E}.$$

We consider the question whether these embeddings are reversible, and show that (Corollary 6.16, Theorem 6.9)

$$\mathcal{B}^X \not\hookrightarrow \mathcal{K}_2^X \not\hookrightarrow \mathcal{K}_1^X$$

and that $\mathcal{E}$ does not embed in $\mathcal{K}_2^{\text{eff}}$ (Corollary 7.14), and hence also not in $\mathcal{K}_1$. We also show that the pca of λ-calculus up to β or $\beta\eta$-equivalence is not embeddable in $\mathcal{K}_1$ (Corollary 5.2).

Aside from comparing the embeddings between relativizations of different models, we also study the embeddings between different relativizations of the same model, and give necessary and sufficient conditions for when $\mathcal{K}_2^X \hookrightarrow \mathcal{K}_2^Y$ (Theorem 6.6), $\mathcal{B}^X \hookrightarrow \mathcal{B}^Y$ (Theorem 6.12), and $\mathcal{G}^X \hookrightarrow \mathcal{G}^Y$ (Theorem 7.10).

Our notation is mostly standard. For pcas our notation follows van Oosten [18] and for computability theory Odifreddi [16]. We use calligraphic letters ($\mathcal{A}$, $\mathcal{B}$, ...) for arbitrary pcas and Roman letters for the fixed elements of a pca. We use Roman letters (f, g, ...) for total functions and Greek letters (θ, ρ, ξ, ψ, ...) for partial functions, except for ω which denotes the natural numbers. $\langle\cdot,\cdot\rangle$ denotes an effective pairing scheme satisfying $\langle 0,0\rangle = 0$. D_u denotes the finite set with canonical code u. φ_e denotes the e-th partial computable (p.c.) function. We use Φ_e to denote the e-th Turing functional and Ψ_e to denote the e-th enumeration operator. We use K to denote the halting set, consisting of those e such that $\varphi_e(e)\downarrow$, and for $X \subseteq \omega$ we use X' to denote the halting set relativized to X.

2. Preliminaries

Definition 2.1. A *partial applicative structure* is a set A with a partial binary operation $\cdot$.

We will denote partial applicative structures by calligraphic letters ($\mathcal{A}$, $\mathcal{B}$...). Given $\mathcal{A} = (A,\cdot)$ we will simply write $a \in \mathcal{A}$ for $a \in A$ and ab for $a \cdot b$ when no confusion can occur. The application operator $\cdot$ associates to the left; that is, we will take abc to mean $(ab)c$. We write $t\downarrow$ do denote that all applications in the expression t are defined, and we use Kleene equality $t \simeq s$ to denote that either the expressions t and s are both defined and equal, or both undefined.

A partial combinatory algebra is a partial applicative structure that is combinatory complete, i.e. in which every term is represented by an element of the structure. By a result of Feferman [7], this is equivalent to the following definition.

Definition 2.2. A *partial combinatory algebra* is a partial applicative structure $\mathcal{A}$ which contains distinct elements s and k such that for all $a, b, c \in \mathcal{A}$,

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

The prototypical example of a partial combinatory algebra is Kleene's first model $\mathcal{K}_1 = (\omega, \cdot)$ where $n \cdot m = \varphi_n(m)$.

A pca with a total application operator is called a *combinatory algebra* (ca), and we say that such a pca is *total*.

Partial combinatory algebras can represent the natural numbers and all partial computable functions on them, as well as a pairing function p and projections p_1 and p_2. In particular, the combinators s and k are sufficient to represent the Church numerals $\overline{n}$, which satisfy the property that $\overline{0}fa = a$ and $\overline{n+1}fa = \overline{n}f(fa)$. For a more complete introduction to partial combinatory algebras, we refer the reader to van Oosten [18].

Definition 2.3. Given pcas $\mathcal{A} = (A, \cdot_A)$ and $\mathcal{B} = (B, \cdot_B)$, an injection $f : A \to B$ is an *embedding* if for all $a, a' \in \mathcal{A}$, if $aa'\downarrow$ then $f(a)f(a')\downarrow = f(aa')$. If $\mathcal{A}$ embeds into $\mathcal{B}$ in this way we write $\mathcal{A} \hookrightarrow \mathcal{B}$.

This notion of embedding is the one studied by Bethke [4] and by Asperti and Ciabattoni [1] in the context of completions. It was also used in Shafer and Terwijn [22] in connection with the ordinal analysis of pcas. A stronger notion has been studied by Bethke, Klop, and de Vrijer [5], who required that the elements s and k in $\mathcal{A}$ are sent to their counterparts in $\mathcal{B}$. Conversely, a weaker notion can be obtained by merely requiring the map $\gamma(a) = \{f(a)\}$ to be a partial applicative morphism $\mathcal{A} \to \mathcal{B}$ (cf. Longley and Normann [12]). In contrast to most prior literature, our focus is on embeddings in general, rather than exclusively on completions, which are the special case when $\mathcal{B}$ is a combinatory algebra.

3. Partial numberings and deciders

Our key insight for generalizing Theorems 1.1 and 1.2 is that these theorems make extensive use of the fact that both $\mathcal{K}_1^X$ and $\mathcal{K}_2^X$ can be coded using natural numbers. In [22], Shafer and Terwijn implicitly use the identity coding for $\mathcal{K}_1^X$ and use the partial map $n \mapsto \Phi_n^X$ as a coding of $\mathcal{K}_2^X$. To make these constructions explicit, we will generalize Ershov's notion of a numbering from [6] to permit partiality.

Definition 3.1. Given a set S, a *partial numbering* of S is a surjective partial function $\gamma : \omega \to S$.

If $\gamma(n) = a$ we say that n is a *γ-code of a*. For $n, m \in \mathrm{dom}(\gamma)$ we write $n \not\sim_\gamma m$ to indicate $\gamma(n) \neq \gamma(m)$.

Note that a numbering is not required to be computable in any sense. On the contrary, we define our notions of computability with respect to a choice of numbering. For instance, the embedding from Theorem 1.2 is said to be X-computable. This can be understood as follows:

Definition 3.2. Let S and T be sets equipped with partial numberings $\gamma : \omega \to S$ and $\delta : \omega \to T$. A (partial) function $\alpha : S \to T$ is *(partial) X-computable with respect to γ and δ* if there exists a (partial) X-computable $\beta : \omega \to \omega$ such that for all $n \in \mathrm{dom}(\gamma)$, if $\beta(\gamma(n))\downarrow$ then

$$\beta(\gamma(n)) = \delta(\alpha(n)).$$

We extend this notion of X-computability to n-ary functions in the straightforward way. When the partial numberings are clear from context, we will omit them.

We also generalize the notion of X-c.e. relations to a partially numbered set.

Definition 3.3. Let $X \subseteq \omega$ and S a set with a partial numbering $\gamma : \omega \to S$. We say that a relation R on $\mathrm{dom}(\gamma)^k$ is *X-c.e. with respect to* γ if there exists an X-c.e. relation R' on ω^k such that $R' \restriction \mathrm{dom}(\gamma)^k = R$.

In particular, we say that $\gamma : \omega \to S$ *has X-c.e. inequivalence* if the relation $a \neq b$ on S^2 is X-c.e. with respect to γ.

Given a pca $\mathcal{A}$ and a partial numbering $\gamma : \omega \to \mathcal{A}$, it is natural to ask what the complexity of the application function is with respect to γ.

Definition 3.4. Let $\mathcal{A}$ be a pca with a partial numbering $\gamma : \omega \to \mathcal{A}$ and let $X \subseteq \omega$. We say that γ is *X-effectively pca-valued* if the application operator is partial X-computable; that is, if there exists a partial X-computable function ψ such that for all $n, m \in \mathrm{dom}(\gamma)$, if $\gamma(n)\gamma(m)\downarrow$ then

$$\gamma(n)\gamma(m) = \gamma(\psi(n, m)).$$

We say that ψ γ-represents the application in $\mathcal{A}$.

When X is the empty set, we say that γ is *effectively pca-valued.*

Note that it can happen that for $n, m, k, l \in \mathrm{dom}(\gamma)$, $\gamma(\psi(n, m)) = \gamma(\psi(k, l))$ but $\psi(n, m) \neq \psi(k, l)$.

Definition 3.5. Let $X \subseteq \omega$ and let $\mathcal{A}$ be a pca with an X-effectively pca-valued partial numbering γ. Let ψ γ-represent the application in $\mathcal{A}$. We say γ is a *strongly X-effectively pca-valued* partial numbering if for all $n, m, k, l \in \mathrm{dom}(\gamma)$, if $\gamma(n)\gamma(m)$ and $\gamma(k)\gamma(l)$ are both defined and equal then $\psi(n, m) = \psi(k, l)$.

We say that ψ strongly γ-represents the application in $\mathcal{A}$. Given such a ψ, we define a section $\eta : \mathcal{A} \to \omega$ of γ as follows. Since $\mathcal{A}$ is a pca, there is an element i such that for all $a \in \mathcal{A}$, $\mathrm{i}a = a$. Fixing a γ-code c of i and taking $a \in \mathcal{A}$, we let $\eta(a) = \psi(c, n)$ for some γ-code n of a. Note that since ψ strongly γ-represents the application in $\mathcal{A}$, the choice of the code n does not matter.

Our goal in representing the application is primarily to construct Turing reductions. To this end, we generalize the notion of a characteristic function as follows.

Definition 3.6. Let $X \subseteq \omega$, $\mathcal{A}$ a pca, and $a_\bot, a_\top \in \mathcal{A}$. Given a partial numbering $\gamma : \omega \to \mathcal{A}$, we say that a function $d : \omega \to \omega$ is an $(a_\bot, a_\top)$-*decider for* X if for all $n \in \omega$,

$$n \notin X \Rightarrow \gamma(d(n)) = a_\bot$$
$$n \in X \Rightarrow \gamma(d(n)) = a_\top.$$

The definition is motivated by the following theorem.

Theorem 3.7. *Let $X, Y \subseteq \omega$, $\mathcal{A}$ a pca, $a_\bot \neq a_\top \in \mathcal{A}$, $\gamma : \omega \to \mathcal{A}$ a partial numbering, and let d be a Y-computable $(a_\bot, a_\top)$-decider for X. If γ has Y-c.e. inequivalence, then $X \leqslant_T Y$.*

Proof. Choose $c_\bot, c_\top \in \omega$ to be γ-codes of $a_\bot$ and $a_\top$ respectively. Let R be the Y-c.e. relation such that $R \restriction \mathrm{dom}(\gamma)^2 = \not\sim_\gamma$. To decide whether $n \in X$, compute $d(n)$ and enumerate R until either we discover $c_\bot R d(n)$ or $c_\top R d(n)$. One must be the case, since $c_\bot \not\sim_\gamma c_\top$, and since $c_\bot, c_\top, d(n) \in \mathrm{dom}(\gamma)$. □

An inspection of the above proof shows that the condition that γ has Y-c.e. inequivalence is stronger than required: it suffices for the relations $x \neq a_\bot$ and $x \neq a_\top$ to be Y-c.e. with respect to γ. This strengthening is not required for our results, but we will look at another generalization in sections 6 and 7.

To conclude the preliminaries, let us consider a more general notion of deciders.

Definition 3.8. Let $X \subseteq \omega$, $\mathcal{A}$ a pca, and let $A_\bot$ and $A_\top$ be subsets of $\mathcal{A}$. Let $\gamma : \omega \to \mathcal{A}$ be a partial numbering. We say that $d : \omega \to \omega$ is an $(A_\bot, A_\top)$-decider for X if for all $n \in \omega$,

$$n \notin X \Rightarrow \gamma(d(n)) \in A_\bot$$
$$n \in X \Rightarrow \gamma(d(n)) \in A_\top.$$

Theorem 3.9. *Let $X, Y \subseteq \omega$, $\mathcal{A}$ a pca, and let $A_\bot$ and $A_\top$ be disjoint subsets of $\mathcal{A}$. Let $\gamma : \omega \to \mathcal{A}$ be a numbering and let d be a Y-computable $(a_\bot, a_\top)$-decider for X. If γ has Y-c.e. inequivalence and there exist Y-c.e. sets $C_\bot$ and $C_\top$ such that $\gamma(C_\bot) = A_\bot$ and $\gamma(C_\top) = A_\top$, then $X \leqslant_T Y$.*

Proof. The proof is analogous to Theorem 3.7; to decide $n \in X$, enumerate $c_\bot \in C_\bot$ and $c_\top \in C_\top$ and wait until we see $d(n) \not\sim_\gamma c_\bot$ or $d(n) \not\sim_\gamma c_\top$. □

4. Kleene's first model

Let us now consider the example of $\mathcal{K}_1^X$, the pca structure on ω where $n \cdot m = \Phi_n^X(m)$. For every $X \subseteq \omega$, the identity is a strongly X-effectively pca-valued partial numbering $\omega \to \mathcal{K}_1^X$, since the application is itself partial X-computable.

In [22], Terwijn and Shafer have shown that an embedding $\mathcal{K}_1^X \to \mathcal{K}_1^Y$ exists if and only if $X \leqslant_T Y$. Our new terminology gives rise to another, more general, proof of each direction. Let us begin with the 'only if' direction:

Theorem 4.1. *Let $X, Y \subseteq \omega$, $\mathcal{A}$ a pca, and $\gamma : \omega \to \mathcal{A}$ a Y-effectively pca-valued partial numbering. Given an embedding $f : \mathcal{K}_1^X \hookrightarrow \mathcal{A}$, there exists a Y-computable $(a_\bot, a_\top)$-decider for X for any choice of $a_\bot, a_\top \in \mathrm{ran}(f)$.*

First, let us see that this does indeed imply the aforementioned 'only if' direction.

Corollary 4.2. *Let $X, Y \subseteq \omega$, $\mathcal{A}$ a pca, and $\gamma : \omega \to \mathcal{A}$ a Y-effectively pca-valued partial numbering. If γ has Y-c.e. inequivalence and there exists an embedding $f : \mathcal{K}_1^X \hookrightarrow \mathcal{A}$ then $X \leqslant_T Y$.*

Proof. This is an immediate consequence of Theorems 4.1 and 3.7. □

Corollary 4.3. *If there exists an embedding $f : \mathcal{K}_1^X \hookrightarrow \mathcal{K}_1^Y$, then $X \leqslant_T Y$.*

Proof. Take $\mathcal{A} = \mathcal{K}_1^Y$ and let γ be the identity seen as a Y-effectively pca-valued partial numbering. Since inequivalence is simply inequality, it is decidable and hence also Y-c.e. The conclusion follows by Corollary 4.2. □

Proof of Theorem 4.1. Let e be a code of the characteristic function of X. Let $x_\perp, x_\top \in \omega$ be such that $f(x_\perp) = a_\perp$ and $f(x_\top) = a_\top$. By the S-m-n theorem there exist partial X-computable functions t and h such that for all $i, j \in \omega$,

$$\Phi^X_{t(i)}(j) = \Phi^X_i(j+1)$$

$$h(i) = \begin{cases} x_\perp & \text{if } \Phi^X_i(0) = 0 \\ x_\top & \text{if } \Phi^X_i(0) = 1 \\ \uparrow & \text{otherwise} \end{cases}.$$

Since these functions are X-computable, they have codes $c_t, c_h \in \omega$.

For every $n \in \omega$, the term $c_h \cdot (\overline{n} \cdot c_t \cdot e)$ is equal to $x_\perp$ if $n \notin X$ and to $x_\top$ otherwise. This can be seen as follows: by induction on n we have $\Phi^X_{\overline{n} \cdot c_t \cdot e}(x) = \Phi^X_e(x+n)$. It follows that by definition of h,

$$c_h \cdot (\overline{n} \cdot c_t \cdot e) = h(\overline{n} \cdot c_t \cdot e) = \begin{cases} x_\perp & \text{if } \Phi^X_e(n) = 0 \\ x_\top & \text{if } \Phi^X_e(n) = 1 \\ \uparrow & \text{otherwise} \end{cases}.$$

Since Φ^X_e is a characteristic function it is total and $(0,1)$-valued, hence the third case never occurs.

It remains to construct d. Note that $c_h \cdot (\overline{n} \cdot c_t \cdot e)$ is a term in $\mathcal{K}_1^X$, but d must give us γ-codes of elements in $\mathcal{A}$. We now use the embedding $f : \mathcal{K}_1^X \hookrightarrow \mathcal{A}$. Since f respects application, we can write $f(c_h \cdot (\overline{n} \cdot c_t \cdot e))$ as $f(c_h) \cdot (f(\overline{n}) \cdot f(c_t) \cdot f(e))$. Fixing γ-codes of $f(c_h)$, $f(c_t)$, $f(e)$, $f(\mathrm{s})$, and $f(\mathrm{k})$ we can thus use ψ, the γ-representation of application in $\mathcal{A}$, to construct a γ-code $d(n)$ for $f(c_h \cdot (\overline{n} \cdot c_t \cdot e))$.[1] This is indeed Y-computable, since ψ is a partial Y-computable function, and d is total since $c_h \cdot (\overline{n} \cdot c_t \cdot e)\downarrow$.

We thus have

$$\gamma(d(n)) = f(c_h \cdot (\overline{n} \cdot c_t \cdot e)) = \begin{cases} f(x_\perp) = a_\perp & \text{if } n \notin X \\ f(x_\top) = a_\top & \text{if } n \in X \end{cases},$$

as desired. □

[1] Note that $f(\overline{n})$ can be constructed this way because $\overline{n}$ can be expressed in terms of s and k.

The above proof and corollary contain the essential themes that will come back repeatedly in this paper. When our goal is to show that an embedding $f : \mathcal{A}' \to \mathcal{A}$ gives rise to a reduction from X to Y, we will find a representation e of X in $\mathcal{A}'$ and then use two carefully chosen elements t (for *tail*) and h (for *head*), to construct the term $h(\overline{n}te)$ that depends on whether $n \in X$. We will then use a Y-effectively pca-valued numbering $\gamma : \omega \to \mathcal{A}$ to construct a γ-code for the term $f(h(\overline{n}te))$, which can be done Y-computably because it requires us to know the code of only the finitely many γ-codes of $f(h)$, $f(t)$, etc. The resulting function $d : \omega \to \omega$ will then suffice to construct the desired reduction.

Let us now look at the other generalization of Theorem 1.1, where we show that for an arbitrary pca $\mathcal{A}$ equipped with a suitable partial numbering γ we can construct an embedding $\mathcal{A} \hookrightarrow \mathcal{K}_1^X$.

Theorem 4.4. *Let $X \subseteq \omega$ and let $\mathcal{A}$ be a pca with a strongly X-effectively pca-valued partial numbering γ. There exists a computable embedding $f : \mathcal{A} \hookrightarrow \mathcal{K}_1^X$.*

Proof. This proof is analogous to the proof of Theorem 1.1 in [22].

Making use of the S-m-n and recursion theorems, define g to be an injective function satisfying

$$\Phi^X_{g(n)}(m) = \begin{cases} g(\psi(n,k)) & \text{if } \exists k.\, g(k) = m \\ \uparrow & \text{otherwise} \end{cases}.$$

Recall that since γ is strongly X-effectively pca-valued, it has a section η that sends every $a \in \mathcal{A}$ to its chosen γ-code. The embedding f is now given by the composition $g \circ \eta$, which is injective since both g and η are injective.

Let $a, b \in \mathcal{A}$ and suppose $ab\downarrow$. Since $\gamma(\psi(\eta(a), \eta(b))) = \gamma(\eta(ab))$, we have $\psi(\eta(a), \eta(b)) = \eta(ab)$ by the choice of η. It follows that

$$g(\eta(a)) \cdot g(\eta(b)) = \Phi^Y_{g(\eta(a))}(g(\eta(b))) = g(\psi(\eta(a), \eta(b))) = g(\eta(ab))$$

as desired. □

Corollary 4.5. *If $X \leqslant_T Y$ then there exists a computable embedding $\mathcal{K}_1^X \hookrightarrow \mathcal{K}_1^Y$.*

Proof. If $X \leqslant_T Y$, every strongly X-effectively pca-valued partial numbering is also strongly Y-effectively pca-valued. Hence the identity gives us the desired numbering and the existence of the embedding follows from Theorem 4.4. □

5. The λ-calculus

The λ-calculus naturally gives rise to a ca λ where the elements are λ-terms up to $\alpha\beta$-equivalence, and the application operation is given by application of λ-terms. We can quotient this ca by an equivalence relation to obtain a new applicative structure, which will again be a ca as long as it is not trivial.

Let $\lambda\eta$ be the λ-terms up to $\alpha\beta\eta$-equivalence. This is an extensional ca.

Theorem 5.1. *Let $\mathcal{A}$ be a pca with an effectively pca-valued partial numbering $\gamma : \omega \to \mathcal{A}$. Given an embedding $f : \lambda\eta \hookrightarrow \mathcal{A}$ or $f : \lambda \hookrightarrow \mathcal{A}$, there* *exists for some $a \in \mathcal{A}$ a computable $(\{a\}, \mathcal{A} - \{a\})$-decider for the halting set K.*

Proof. Given a natural number n, let $\overline{n}$ be the Church numeral n seen as a λ-term. Let R be a λ-term such that for all $n, m, s \in \omega$ and all λ-terms x, y,

$$R\,\overline{n}\,\overline{m}\,\overline{s}\,x\,y = \begin{cases} x & \text{if } \varphi_{n,s}(m)\downarrow \\ y & \text{otherwise} \end{cases}.$$

Using the fixed point combinator Y, define

$$H = \lambda x.\, Y(\lambda f\, s.\, R\,x\,x\,s\,\overline{0}\,(f\,(s+\overline{1})))(\overline{0}).$$

Recall that K is the halting set, the set of $n \in \omega$ such that $\varphi_n(n)\downarrow$. Note that the term $H\overline{n}$ has a normal form if and only if $n \in K$, and this holds both with respect to λ and $\lambda\eta$. By combinatory completeness, H can be defined using only the s and k combinators. We can thus effectively construct a γ-code $d(n)$ for $f(H\overline{n})$. Since $\gamma(d(n)) = f(\overline{0})$ if and only if $n \in K$, it follows that d is an $\big(\{f(\overline{0})\}, \mathcal{A} - \{f(\overline{0})\}\big)$-decider for K. □

Corollary 5.2. *For any $X \subseteq \omega$, if $\lambda\eta$ or λ embeds in $\mathcal{K}_1^X$ then $K \leqslant_T X$.*

Proof. We use the identity numbering for $\mathcal{K}_1^X$, which is X-effectively pca-valued. It is easy to verify that the conditions for Theorem 3.9 are satisfied. □

This result provides the following corollary: for a pca $\mathcal{A}$, there cannot exist both an embedding $\lambda \hookrightarrow \mathcal{A}$ (respectively $\lambda\eta \hookrightarrow \mathcal{A}$) and an embedding $\mathcal{A} \hookrightarrow \mathcal{K}_1$. It follows that to demonstrate that $\mathcal{A}$ does not embed in $\mathcal{K}_1$, it suffices to show that $\mathcal{A}$ is a faithful model of the λ-calculus (respectively $\lambda\eta$-calculus); that is, a model which does not identify any $\alpha\beta$-inequivalent (respectively $\alpha\beta\eta$-inequivalent) terms.

6. Kleene's second model

We will use the coding of $\mathcal{K}_2$ from Shafer and Terwijn [22], in which application in $\mathcal{K}_2$ is defined for all $g, h \in \omega^\omega$ by

$$g \cdot h = \Phi_{g(0)}^{g \oplus h}, \tag{1}$$

where it is understood that $g \cdot h$ is defined if and only if the function computed on the right is total, and Φ_e denotes the e-th Turing functional, viewed as a function of the oracle. This coding is much easier to work with than Kleene's original definition. In Appendix A we will show that the two codings are equivalent, in the sense that they embed into each other.

Note that if we allow the elements to be partial functions, this coding gives van Oosten's model $\mathcal{B}$ for sequential computations from [19], up to the same differences in coding as above. Since the application is (up to totality) defined the same in both cases, the inclusion is an embedding.

For a set of natural numbers X, let $\mathcal{K}_2^X$ denote the restriction of $\mathcal{K}_2$ to those $g \in \mathcal{K}_2$ that are X-computable. Since s and k in $\mathcal{K}_2$ can be chosen to be computable, $\mathcal{K}_2^X$ is a pca for every X. Similarly, we denote the restriction of $\mathcal{B}$ to the partial X-computable functions by $\mathcal{B}^X$. This too is a pca for every X.

Theorem 6.1. *Let $X \subseteq \omega$ and let $\mathcal{A}$ be a pca with a strongly X-effectively pca-valued partial numbering γ. Then $\mathcal{A}$ is embeddable in $\mathcal{K}_2^X$ and hence in $\mathcal{K}_2$.*

Note that by Theorem 4.4, it would suffice to show that for all $X \subseteq \omega$ there exists an embedding $\mathcal{K}_1^X \hookrightarrow \mathcal{K}_2^X$. However, since the proof is essentially the same as the general case, we choose to present this result directly.

Proof. We define an injective function $f : \mathcal{A} \to \mathcal{K}_2^X$ such that

$$a \cdot b{\downarrow} = c \Longrightarrow f(a) \cdot f(b){\downarrow} = f(c). \tag{2}$$

What makes (2) hard to realize is that $f(a) \cdot f(b)$ has to produce the *same* $f(c) \in \omega^\omega$, for all the different a and b such that $a \cdot b = c$ in $\mathcal{A}$.

Fortunately, the embedding f does not need to be effective, and since $f(a) \in \omega^\omega$ is an infinite object, we can code all the necessary information into $f(a)$ as follows. Recall that η is the section of γ that exists since γ is strongly X-effectively pca-valued. Let ψ be the γ-representation of the application in $\mathcal{A}$ and let ψ_s be the s-step approximation of ψ.

For every $a \in \mathcal{A}$ we define:

$$\begin{aligned} f(a)(0) &= e \text{ (a code defined below)} \\ f(a)(1) &= \eta(a) \\ f(a)(2 + \langle n, m, s\rangle) &= \begin{cases} 1 + \psi_s(n,m) & \text{if } \psi_s(n,m){\downarrow}, \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Note that for every $i \neq 1$, the information $f(a)(i)$ is the same for all a. This information is computable from X, hence $f(a) \in \mathcal{K}_2^X$.

Next we use the recursion theorem to define the code e such that

$$\begin{aligned} &\Phi_e^{g \oplus h}(0) = e \\ &\Phi_e^{g \oplus h}(1) = k \text{ (where } k \text{ is as explained below)} \\ &\Phi_e^{g \oplus h}(i) = g(i) \ (i > 1) \end{aligned}$$

The code k in the second line is computed as follows. First extract the codes $n = g(1)$ and $m = h(1)$. Then search for the least s such that $g(2 + \langle n, m, s\rangle) = 1 + k$, and output k. If k is not found the value is undefined.

Now we observe that if $a \cdot b{\downarrow} = c$ in $\mathcal{A}$, and $n = \eta(a), m = \eta(b), k = \eta(c)$, then

$$f(c)(1) = (f(a) \cdot f(b))(1) = \Phi_e^{f(a) \oplus f(b)}(1) = k,$$

because Φ_e extracts n and m from $f(a)$ and $f(b)$, and then searches for the minimal s such that $f(a)(2 + \langle n, m, s\rangle) = 1 + k$. This shows that (2) is satisfied. □

Corollary 6.2. *Every countable pca $\mathcal{A}$ embeds in $\mathcal{K}_2$.*

Proof. Recall that by definition, every pca is non-empty. It is easy to show that every pca is infinite (cf. van Oosten [18]). Take a bijective function $\gamma : \omega \to \mathcal{A}$ and let $\psi(n,m) = k$ if $\gamma(n)\gamma(m) = \gamma(k)$. Let X be the graph of ψ. By Theorem 6.1, $\mathcal{A}$ embeds in $\mathcal{K}_2^X$ and hence in $\mathcal{K}_2$. □

In order to better understand embeddings between $\mathcal{K}_2^X$ and $\mathcal{K}_2^Y$, we look at embeddings out of $\mathcal{K}_2^X$ into an arbitrary partially numbered pca.

Theorem 6.3. *Let $X, Y \subseteq \omega$, $\mathcal{A}$ a pca, and $\gamma : \omega \to \mathcal{A}$ a Y-effectively pca-valued partial numbering. Given an embedding $f : \mathcal{K}_2^X \hookrightarrow \mathcal{A}$ and any $g_\perp, g_\top \in \mathcal{K}_2^X$, there is a Y-computable $(f(g_\perp), f(g_\top))$-decider d for X.*

Note that the existence of an embedding $f' : \mathcal{K}_1^X \hookrightarrow \mathcal{K}_2^X$ and Theorem 4.1 are together not enough to conclude this, since we may have that $g_\perp$ or $g_\top$ falls outside $\operatorname{ran}(f')$.

Proof. Define $t, h \in \mathcal{K}_2^X$ by

$$(tx)(n) = x(n+1)$$

$$(hx)(n) = \begin{cases} g_\perp(n) & \text{if } x(0) = 0 \\ g_\top(n) & \text{if } x(0) = 1 \\ \uparrow & \text{otherwise} \end{cases}.$$

Note that χ_X, the characteristic function of X, is itself an element of $\mathcal{K}_2^X$. We can thus construct a Y-computable d such that for all $n \in \omega$, $\gamma(d(n)) = f(h(\overline{n} t \chi_X))$. We see that $h(\overline{n} t \chi_X) = g_\perp$ if $n \notin X$ and $h(\overline{n} t \chi_X) = g_\top$ otherwise, as desired. □

As in the case of the λ-calculus, this immediately gives us a result on the embeddability of $\mathcal{K}_2^X$ in $\mathcal{K}_1^Y$. However, we will see in Theorem 6.9 that a more complicated construction allows for a stronger result.

Corollary 6.4. *If there exists an embedding $\mathcal{K}_2^X \hookrightarrow \mathcal{K}_1^Y$ then $X \leqslant_T Y$.*

Proof. This follows directly from Theorems 6.3 and 3.7. □

In order to also say something about embeddings $\mathcal{K}_2^X \hookrightarrow \mathcal{K}_2^Y$, we require a lemma on the complexity of $\not\sim_\gamma$.

Lemma 6.5. *Let $X \subseteq \omega$. The numbering $\gamma(n) = \Phi_n^X$, restricted to those $n \in \omega$ such that $\gamma(n)$ is total, is an effectively pca-valued partial numbering, which we will denote by γ_t. This numbering has X-c.e. inequivalence.*

Proof. Let $n, m \in \omega$ such that $\gamma_t(n)\downarrow, \gamma_t(m)\downarrow$. By the S-m-n theorem, there is a computable function g such that

$$\Phi_{g(n,m)}^X(k) = (\Phi_n^X \cdot \Phi_m^X)(k),$$

since the right-hand side is X-computable. This g γ-represents application in $\mathcal{K}_2^X$, showing that γ_t is indeed an effectively pca-valued partial numbering.

Let R be the binary relation on ω defined by

$$nRm \Leftrightarrow \exists k.\, \exists s.\, \Phi_{n,s}^X(k)\downarrow \wedge \Phi_{m,s}^X(k)\downarrow \wedge \Phi_{n,s}^X(k) \neq \Phi_{m,s}^X(k).$$

This is X-c.e., and $R \restriction \operatorname{dom}(\gamma)^2 = \not\sim_{\gamma_t}$, hence γ_t has X-c.e. inequivalence. □

Theorem 6.6. *Let $X, Y \subseteq \omega$. There exists an embedding $\mathcal{K}_2^X \to \mathcal{K}_2^Y$ if and only if $X \leqslant_T Y$.*

Proof. If $X \leqslant_T Y$ then every X-computable function is in fact Y-computable and hence $\mathcal{K}_2^X$ is a sub-pca of $\mathcal{K}_2^Y$ and the inclusion is an embedding.

Conversely, let $d : \omega \to \omega$ be the decider constructed in Theorem 6.3. By Lemma 6.5, $\mathcal{K}_2^Y$ has a Y-effectively pca-valued partial numbering γ_t, and hence by Theorem 3.7, since inequivalence in $\mathcal{K}_2^Y$ is Y-c.e., we have $X \leqslant_T Y$. □

Theorem 6.7. *Let $X, Y \subseteq \omega$. There exists an embedding $\mathcal{K}_1^X \to \mathcal{K}_2^Y$ if and only if $X \leqslant_T Y$.*

Proof. The "if" direction is simply Theorem 1.2 from Shafer and Terwijn [22], together with the fact that if $X \leqslant_T Y$ then $\mathcal{K}_2^X \hookrightarrow \mathcal{K}_2^Y$ via the inclusion.

By Lemma 6.5, $\mathcal{K}_2^Y$ has a Y-effectively pca-valued partial numbering γ which has Y-c.e. inequivalence. The result follows by Corollary 4.2. □

Let us return to the question of when there is an embedding $\mathcal{K}_2^X \hookrightarrow \mathcal{K}_1^Y$.

Lemma 6.8. *Let $X, Y \subseteq \omega$ and let Z be an X-c.e. set. Let $\mathcal{A}$ be a pca and $\gamma : \omega \to \mathcal{A}$ be a Y-efficiently pca-valued partial numbering. If there exists an embedding $f : \mathcal{K}_2^X \hookrightarrow \mathcal{A}$, then there exists an $a \in \mathcal{A}$ and a Y-computable $(\{a\}, \mathcal{A} - \{a\})$-decider d for Z.*

Proof. To begin, we encode the set X into an element g of $\mathcal{K}_2^X$. We use $g(0)$ to represent a code e, which we will define shortly. Define

$$g(0) = e$$

$$g(n+1) = \begin{cases} 0 & \text{if } n \notin X \\ 1 & \text{if } n \in X \end{cases}.$$

Since Z is X-c.e., it has an s-step X-computable approximation Z_s. We define e to be a code such that for all $h \in \mathcal{K}_2^X$,

$$\Phi_e^{g \oplus h}(s) = \begin{cases} 0 & \text{if } h(0) \notin Z_s \\ 1 & \text{if } h(0) \in Z_s \end{cases}.$$

The family of functions h_n, given by $h_n(0) = n$ and $h_n(k+1) = 0$, can be uniformly constructed out of finitely many elements of $\mathcal{K}_2^X$; hence we can Y-computably find a γ-code for every $f(h_n)$.

Let $d(n)$ be a γ-code of $f(g \cdot h_n)$. Note that $g \cdot h_n$ is the constant zero function exactly when $n \notin Z$. Hence taking a to be the image of the constant zero function under f, we get exactly the desired result. □

Theorem 6.9. *If there is an embedding $\mathcal{K}_2^X \hookrightarrow \mathcal{K}_1^Y$ then $X' \leqslant_T Y$.*

Proof. By Lemma 6.8 there is a Y-computable d and a $c \in \omega$ such that $d(n) = c$ if and only if $n \notin X'$, which suffices. □

Let us now consider the case of $\mathcal{B}$. Note that there is no need to generalize Theorem 6.1 since there is an embedding $\mathcal{K}_2^X \hookrightarrow \mathcal{B}^X$ given by the inclusion, and so the corresponding result is given by composition. We therefore focus on embeddings out of $\mathcal{B}^X$.

Theorem 6.10. *Let $X, Y \subseteq \omega$, $\mathcal{A}$ a pca, and $\gamma : \omega \to \mathcal{A}$ a Y-effectively pca-valued partial numbering. Given an embedding $f : \mathcal{B}^X \hookrightarrow \mathcal{A}$ and any $\theta_\perp, \theta_\top \in \mathcal{B}^X$, there is a Y-computable $(f(\theta_\perp), f(\theta_\top))$-decider d for X.*

Proof. The proof is identical to that of Theorem 6.3; we can perform the same construction, generalizing h to return codes of partial functions. □

The statement corresponding to Theorem 6.6 also holds for $\mathcal{B}$, but the situation is more complicated. The map $\gamma(n) = \Phi_n^X$ is an effectively pca-valued partial numbering, but the relation $\not\sim_\gamma$ for $\mathcal{B}^X$ is not X-c.e.: given codes of $\theta \subseteq \rho \in \mathcal{B}^X$ we cannot tell whether $\mathrm{dom}(\theta) = \mathrm{dom}(\rho)$. In order to avoid this problem we must ensure that we find a decider whose image consists of two incomparable elements. Fortunately, this is always possible.

Lemma 6.11. *Let $\mathcal{A}$ and $\mathcal{A}'$ be pcas and let $f : \mathcal{A} \hookrightarrow \mathcal{A}'$ be an embedding. If $\leqslant$ is a partial order on $\mathcal{A}'$ under which the application is monotonic, then* $\mathrm{ran}(f)$ *contains incomparable elements.*

Proof. Recall that every pca has a pairing function p and projections p_1 and p_2. Suppose for a contradiction, without loss of generality, that $f(\mathrm{p}_1) \subset f(\mathrm{p}_2)$. Let $a = \mathrm{p} \cdot \mathrm{p}_2 \cdot \mathrm{p}_1$, then

$$f(\mathrm{p}_2) = f(\mathrm{p}_1 a) = f(\mathrm{p}_1)f(a) \subseteq f(\mathrm{p}_2)f(a) = f(\mathrm{p}_2 a) = f(\mathrm{p}_1),$$

giving a contradiction. □

Theorem 6.12. *Let $X, Y \subseteq \omega$. There exists an embedding $\mathcal{B}^X \to \mathcal{B}^Y$ if and only if $X \leqslant_T Y$.*

Proof. One direction is trivial, as in the case of Theorem 6.6.

For the other direction, let $\gamma : \omega \to \mathcal{B}^Y$ be the mapping $\gamma(n) = \Phi_n^Y$. This is a Y-effectively pca-valued partial numbering, with the construction being the same as in Lemma 6.5.

By Theorem 6.10 there is an $(f(\theta_\perp), f(\theta_\top))$-decider d for X, where $f(\theta_\perp)$ and $f(\theta_\top)$ can be taken to be incomparable. Pick a point $k_\perp$ such that $f(\theta_\perp)(k_\perp)\downarrow$ and either $f(\theta_\top)(k_\perp)\uparrow$ or $f(\theta_\perp)(k_\perp) \neq f(\theta_\top)(k_\perp)$, and a point $k_\top$ where the reverse is true. By computing $\Phi_{d(n)}^Y(k_\perp)$ and $\Phi_{d(n)}^Y(k_\top)$, one of which terminates, and comparing it to $f(\theta_\perp)(k_\perp)$ or $f(\theta_\top)(k_\top)$ respectively, we can decide whether $n \in X$. □

Theorem 6.13. *Let $X, Y \subseteq \omega$. There exists an embedding $\mathcal{K}_2^X \to \mathcal{B}^Y$ if and only if $X \leqslant_T Y$.*

Proof. One direction is trivial, since if $X \leqslant_T Y$ then $\mathcal{K}_2^X$ is a sub-pca of $\mathcal{B}^Y$.

For the other direction, let $f : \mathcal{K}_2^X \to \mathcal{B}^Y$ be an embedding and pick $g_\perp, g_\top \in \mathcal{K}_2^X$ such that $f(g_\perp)$ and $f(g_\top)$ are incomparable; this is possible by Lemma 6.11.

Now let $d : \omega \to \omega$ be the Y-computable $(f(g_\perp), f(g_\top))$-decider given by Theorem 6.3. Since $f(g_\perp)$ and $f(g_\top)$ are incomparable, we can Y-computably decide whether $k \in X$ by enumerating the graphs of $f(g_\perp)$ and $f(g_\top)$ until we see these disagree, as we do in the proof of Theorem 6.12. □

Let us now show that embeddings out of $\mathcal{B}^X$ can allow us to construct deciders for more than just X. Compare this to Theorems 6.9 and 7.13.

Theorem 6.14. *Let $X, Y \subseteq \omega$ and let $\mathcal{A}$ be a pca with a Y-effectively pca-valued partial numbering γ. Given an embedding $f : \mathcal{B}^X \hookrightarrow \mathcal{A}$ and any $\theta \in \mathcal{B}^X$, there is a Y-computable $(f(\emptyset), f(\theta))$-decider d for every X-c.e. set Z.*

Proof. The proof is similar to that of Theorem 6.3. Let Z be an X-c.e. set and let ξ be an X-computable partial function such that $\mathrm{dom}(\xi) = Z$.

Define $t, h \in \mathcal{B}^X$ by

$$t(\mu)(n) = \mu(n+1)$$
$$h(\mu)(n) = \begin{cases} \theta(n) & \text{if } \mu(0)\downarrow \\ \uparrow & \text{otherwise} \end{cases}.$$

It may be tempting to think that we can take any $\theta_\perp \subseteq \theta_\top$ instead of $\emptyset$ and θ, but this is not the case, since we cannot perform the computations $\mu(0)$ and $\theta_\perp(n)$ in parallel.

As in the proof of Theorem 6.3, the term $h(\overline{n}t\xi)$ is $\emptyset$ if $n \notin X$ and is θ if $n \in X$. Letting $d(n)$ be a code of $f(h(\overline{n}t\xi))$, which can be constructed Y-computably using the fact γ is a Y-effectively pca-valued partial numbering, we see that d is indeed an $(f(\emptyset), f(\theta))$-decider for X'. □

Corollary 6.15. *Let $X, Y \subseteq \omega$. If there exists an embedding $f : \mathcal{B}^X \hookrightarrow \mathcal{K}_2^Y$ then $X' \leqslant_T Y$.*

Proof. This follows from Theorems 6.14 and 3.7 since X' is X-c.e. □

Corollary 6.16. *There is no embedding $\mathcal{B}^X \hookrightarrow \mathcal{K}_2^X$ for any X.*

Since $\mathcal{K}_1^X$ embeds in $\mathcal{K}_2^X$, Corollary 6.16 holds if we replace $\mathcal{K}_2^X$ with $\mathcal{K}_1^X$. However, in this case we can say something more precise.

Theorem 6.17. *For every $X \subseteq \omega$, there is an embedding $\mathcal{B}^X \hookrightarrow \mathcal{K}_1^{X''}$.*

Proof. Let $X \subseteq \omega$. There is a total computable binary function h such that in $\mathcal{B}^X$ we have for all n, m,

$$\Phi^X_{h(n,m)} = \Phi^X_n \cdot \Phi^X_m.$$

Note that using X'', we can decide, given n and m, whether $\Phi^X_n = \Phi^X_m$. Let h' be the X''-computable function such that for all n, m, $\Phi^X_{h(n,m)} = \Phi^X_{h'(n,m)}$ and every $h'(n,m)$ is minimal. Using the S-m-n and recursion theorems, define g to be an injective computable function such that

$$\Phi^{X''}_{g(n)}(k) = \begin{cases} g(h'(n,m)) & \text{if } m \text{ is minimal such that } g(m) = k \\ \uparrow & \text{otherwise.} \end{cases}$$

We are now ready to define our embedding $f : \mathcal{B}^X \hookrightarrow \mathcal{K}_1^{X''}$. Given a $\psi \in \mathcal{B}^X$, let $f(\psi) = g(n)$ where n is minimal such that $\Phi^X_n = \psi$.

To verify that this construction is correct, let Φ^X_n, Φ^X_m be elements of $\mathcal{B}^X$ with n, m minimal. By construction, $h'(n,m)$ is the minimal code of $\Phi^X_n \cdot \Phi^X_m$. Now in $\mathcal{K}_1^{X''}$ we have

$$g(n) \cdot g(m) = \Phi^{X''}_{g(n)}(g(m)) = g(h'(n,m)),$$

as desired. □

Theorem 6.18. *For all $X, Y \subseteq \omega$, if there exists an embedding $f : \mathcal{B}^X \hookrightarrow \mathcal{K}_1^Y$ then $X'' \leqslant_T Y$.*

Proof. We show that the Π_2^X-complete set

$$\{\langle n,m\rangle \mid \Phi_n^X = \Phi_m^X\}$$

is Y-computable.

There is a finite set $E \subseteq \mathcal{B}^X$ such that every $\Phi_n^X \in \mathcal{B}^X$ can be written as a term consisting of elements of E, where a description of this term is computable from n.[2]

Suppose now that there exists an embedding $f : \mathcal{B}^X \hookrightarrow \mathcal{K}_1^Y$. For every element $\psi \in E$ we have a corresponding code $f(\psi) \in \mathcal{K}_1^Y$. Since E is finite, we can thus construct a Y-computable function g such that $g(n) = f(\Phi_n^X)$. Now, since f is an embedding, for any $n, m \in \omega$, $g(n) = g(m)$ if and only if $\Phi_n^X = \Phi_m^X$, giving the desired result. □

Together, Theorems 6.17 and 6.18 precisely characterize the embeddings $\mathcal{B}^X \hookrightarrow \mathcal{K}_1^Y$, since by Theorem 1.1 $Z \leqslant_T Y$ implies that $\mathcal{K}_1^Z$ embeds in $\mathcal{K}_1^Y$.

Corollary 6.19. *For all $X, Y \subseteq \omega$, there exists an embedding $\mathcal{B}^X \hookrightarrow \mathcal{K}_1^Y$ if and only if $X'' \leqslant_T Y$.*

Note that Theorem 6.18 can be seen as analogous to Theorem 6.9 and Corollary 6.15. We expect Theorem 6.17 to also have such analogous results.

Conjecture 6.20. *For every $X \subseteq \omega$, there exist embeddings $\mathcal{K}_2^X \hookrightarrow \mathcal{K}_1^{X'}$ and $\mathcal{B}^X \hookrightarrow \mathcal{K}_2^{X'}$.*

Note that Theorem 6.17 follows from this conjecture, since we can compose the embeddings. All in all, this is an interesting example of a case where working with all partial computable functions is considerably simpler than working only with the total computable functions.

To conclude, take $X \subseteq \omega$ and let us remark on the curious relation between $\mathcal{K}_2^X$ and $\mathcal{B}^X$ that these results reveal. We know that $\mathcal{K}_2^X$ embeds in $\mathcal{B}^X$ (Theorem 6.13) but not vice-versa (Corollary 6.15). This suggests that $\mathcal{B}^X$ is in some sense 'more powerful' than $\mathcal{K}_2^X$. Also note that $\mathcal{K}_2^X$ is never isomorphic to $\mathcal{B}^Y$ for any Y (in whatever sense of the notion of isomorphic that preserves embeddability). Namely, $\mathcal{K}_2^X$ and $\mathcal{B}^Y$ never embed into the same pcas: If $X \leqslant_T Y$ then $\mathcal{K}_2^X \hookrightarrow \mathcal{K}_2^Y$ but $\mathcal{B}^Y \not\hookrightarrow \mathcal{K}_2^Y$, and if $X \not\leqslant_T Y$ then $\mathcal{B}^Y \hookrightarrow \mathcal{B}^Y$ but $\mathcal{K}_2^X \not\hookrightarrow \mathcal{B}^Y$.

7. Scott's graph model

In 1969, Dana Scott introduced his graph model [21], which gives rise to the following pca structure on $\mathcal{P}(\omega)$.

Let D_u denote the canonical finite set with code u. We follow the convention that for finite sets $X \subset \omega$, X is coded by $\sum_{k\in X} 2^k$. In particular, for every $n \in \omega$, $D_{2^n} = \{n\}$. Let $\langle\cdot,\cdot\rangle$ denote a bijective pairing scheme such that $\langle 0, 0\rangle = 0$.

We define an application operator on $\mathcal{P}(\omega)$ as follows: given $X, Y \subseteq \omega$, let

$$X \cdot Y = \{n \mid \exists u.\, \langle n,u\rangle \in X \wedge D_u \subseteq Y\}. \tag{3}$$

[2] To see how to achieve this, let $h_n(x) = n$ and choose $j \in \mathcal{B}^X$ such that $j \cdot h_n = \Phi_n^X$. Note that we can build every h_n from a finite set by using the Church encoding of numerals, which in turn can be constructed from only s and k.

This application on $\mathcal{P}(\omega)$ gives rise to a pca that we will denote by $\mathcal{G}$. A proof that $\mathcal{G}$ is indeed a pca can be found in Scott [21].[3]

Following Soare [23], we will use $\mathcal{E}$ to denote the class of c.e. sets. For any set $X \subseteq \omega$ we will use $\mathcal{G}^X$ to denote the least class of sets that contains X and all c.e. sets and is closed under application. Note that this is not the same as the X-c.e. sets; for example, for any c.e. set X, $\mathcal{G}^X = \mathcal{E}$. For every X, $\mathcal{G}^X$ with the application (3) is again a pca, since the combinators s and k of $\mathcal{G}$ can be taken in $\mathcal{E}$ (cf. Scott [21].)

The application operator (3) and the pcas $\mathcal{G}^X$ are closely connected with the structure of enumeration degrees (cf. Odifreddi [17, Chapter XIV]). Recall that X is *e-reducible* to Y ($X \leqslant_e Y$) if for some c.e. relation R,

$$x \in X \Leftrightarrow \exists u (R(x,u) \wedge D_u \subseteq Y).$$

Note that this is the case exactly when for some c.e. set A, $A \cdot Y = X$. The set $\mathcal{G}^Y$ thus contains exactly the sets X that are e-reducible to Y. This connection is further discussed in Scott [21].

Embeddings into graph models have previously been studied by Bethke in [4, Chapter 6], but the proof employed a different graph pca: given a set A, Bethke (cf. [4, Chapter 3, Definition 0.2]) defines a family of sets

$$\begin{aligned} G_0(A) &= A \\ G_{n+1}(A) &= G_n(A) \cup \{(B,b) \mid B \in \mathcal{P}^{\mathrm{fin}}(G_n(A)), b \in G_n(A)\} \\ G(A) &= \bigcup_{n\in\omega} G_n(A) \end{aligned}$$

and defines the application operator on $\mathcal{P}(G(A))$ by

$$X \cdot Y = \{b \mid \exists B \subseteq Y.\, (B,b) \in X\}.$$

This gives rise to a pca, and Bethke showed that every pca $\mathcal{A} = (A, \cdot)$ embeds into $\mathcal{P}(G(A))$. In contrast, our result shows that countable pcas can be embedded into Scott's graph model $\mathcal{G}$.

Theorem 7.1. *Let $\mathcal{A}$ be a pca with a partial numbering $\gamma : \omega \to \mathcal{A}$ and let*

$$X = \big\{\langle n,m,k\rangle \mid n,m,k \in \omega \wedge \gamma(n)\cdot\gamma(m) = \gamma(k)\big\}. \tag{4}$$

Then $\mathcal{A}$ is embeddable in $\mathcal{G}^X$ and hence in $\mathcal{G}$.

Proof. We construct an embedding $f : \mathcal{A} \to \mathcal{G}^X$. Recall that $D_{2^n} = \{n\}$ in our coding. For every $n \in \omega$, let $n^* = \langle n, 1\rangle$. We will ensure that for all $a \in \mathcal{A}$, $0 \notin f(a)$ and for all $n \in \omega$, if $\gamma(n) = a$ then $n^* \in f(a)$.

We define an $(\mathcal{A}, \omega)$-indexed family of sets $Y_{a,i}$:

$$\begin{aligned} Y_{a,0} &= \{n^* \mid \gamma(n) = a\} \\ Y_{a,i+1} &= \big\{\langle y, 2^{n^*}\rangle \mid n \in \omega, y \in Y_{a\cdot\gamma(n),i}\big\} \end{aligned}$$

For $a \in \mathcal{A}$, let $f(a) = \bigcup_{i\in\omega} Y_{a,i}$.

Recall that every pca contains an element i such that $ia = a$ for all a. By fixing a code e such that $\gamma(e) = \mathrm{i}$, we can use X to enumerate all pairs n, k

[3] Scott does not use the terminology of pcas, but all the ingredients are there. Theorem 2.3 (p173) is the combinatory completeness of $\mathcal{G}$, and by Theorem 2.6 the combinators k and s can be defined using c.e. sets.

such that $\gamma(n) = \gamma(k)$, hence $Y_{a,0} \leqslant_e X$. In particular, we can find a code for such an enumeration effectively from a code for a. By induction, we can use X to effectively find a code for $Y_{a,i}$ for every $a \in \mathcal{A}$ and $i \in \omega$, hence $f(a) \leqslant_e X$.

We will now verify that this construction works. For every $a \in \mathcal{A}$, every element of $f(a)$ is of the form $n^* = \langle 1, n\rangle$ with $\gamma(n) = a$ or $\langle y, 2^{n^*}\rangle$ for some $n, y \in \omega$. These are all tuples unequal to $\langle 0, 0\rangle$, hence $0 \notin f(a)$. It follows that $f(a) \cdot f(b) = f(a) \cdot \{n^* \mid n \in \gamma^{-1}(a)\}$, since all elements of $Y_{b,i}$ for $i > 0$ are not of the form n^*.

Let $a, b, c \in \mathcal{A}$ such that $a \cdot b = c$. By construction, the following are equivalent:

$$\begin{aligned} &y \in f(a) \cdot \{n^* \mid n \in \gamma^{-1}(b)\} \\ \exists i, n \in \omega.\, &\gamma(n) = b \wedge \langle y, 2^{n^*}\rangle \in Y_{a,i+1} \\ \exists i, n \in \omega.\, &\gamma(n) = b \wedge y \in Y_{a\cdot\gamma(n),i} = Y_{a\cdot b,i} = Y_{c,i} \\ &y \in f(c), \end{aligned}$$

as desired. □

For the special case of $\mathcal{E} = \mathcal{G}^{\emptyset}$, we in particular get the following result.

Corollary 7.2. *$\mathcal{K}_1$ is embeddable into $\mathcal{E}$.*

Proof. Taking

$$X = \{\langle n, m, k\rangle \mid n, m, k \in \omega \wedge \varphi(m) = k\},$$

by the theorem, there exists an embedding $\mathcal{K}_1 \to \mathcal{G}^X$. Since X is c.e., $\mathcal{G}^X = \mathcal{E}$, as desired. □

Since we know that $\mathcal{K}_1 \hookrightarrow \mathcal{K}_2^{\text{eff}} \hookrightarrow \mathcal{B}^{\text{eff}}$, Corollary 7.2 can also be obtained using the following, more powerful, result.

Theorem 7.3. *There is an embedding $\mathcal{B} \hookrightarrow \mathcal{G}$, which restricts to an embedding $\mathcal{B}^X \hookrightarrow \mathcal{G}^{X \oplus \overline{X}}$ for all $X \subseteq \omega$.*

Proof. We perform a construction similar to the one above, but use the fact that elements of $\mathcal{B}$ can be approximated by finite sequences.

We construct an embedding $f : \mathcal{B} \hookrightarrow \mathcal{G}$. Given $n \in \omega$, let n^* be $\langle n, 1\rangle$. As before, we ensure that $0 \notin f(\theta)$ for all $\theta \in \mathcal{B}$. Given $u \in \omega$, let $\psi(u)(a) = b$ if there is a unique b such that $\langle a, b\rangle^* \in D_u$ and let it be undefined otherwise. Denote the set of codes of finite sets of elements of the form n^* by C, that is,

$$C = \{u \in \omega \mid \forall k \in D_u.\, \exists n.\, k = n^*\}.$$

We first define f in stages on the finite elements[4] of $\mathcal{B}$, and then define the embedding of $\theta \in \mathcal{B}$ as the union of the embeddings of its finite subfunctions.

[4] That is: partial functions with finite domain.

For $\sigma \in \mathcal{B}$ finite and $\theta \in \mathcal{B}$ arbitrary, define

$$\begin{aligned} f_0(\sigma) &= \{\langle a, b\rangle^* \mid \sigma(a) = b\} \\ f_{n+1}(\sigma) &= \{\langle x, u\rangle \mid x \in f_n(\tau) \wedge u \in C \wedge \tau \subseteq \sigma \cdot \psi(u)\} \\ f(\sigma) &= \bigcup_{n\in\omega} f_n(\sigma) \\ f(\theta) &= \bigcup_{\substack{\sigma\subseteq\theta \\ \sigma \text{ finite}}} f(\sigma). \end{aligned}$$

By induction on the index, it is easy to show that if $\sigma \subseteq \tau$ then $f_n(\sigma) \subseteq f_n(\tau)$, hence the last line of the definition is also consistent for finite θ.

Note that f is injective since for all $n, k \in \omega$, $\langle n, k\rangle^* \in f(\theta)$ iff $\theta(n) = k$ allowing us to reconstruct θ from $f(\theta)$. Furthermore, for all $u \in C$ and $\theta \in \mathcal{B}$, if $D_u \subset f(\theta)$ then $\psi(u) \subseteq \theta$.

To verify this construction is correct, let $\theta, \rho \in \mathcal{B}$. Suppose $x \in f(\theta)\cdot f(\rho)$. Then there is some $\langle x, u\rangle \in f(\theta)$ such that $D_u \subseteq f(\rho)$. By construction, $u \in C$ and there are $\sigma \subseteq \theta$ and τ such that $\tau \subseteq \sigma\cdot\psi(u)$ and $x \in f(\tau)$. Since $D_u \subset f(\rho)$ it follows that $\psi(u) \subseteq \rho$ and hence $\tau \subseteq \sigma \cdot \psi(u) \subseteq \theta \cdot \rho$. We thus have $x \in f(\tau) \subseteq f(\theta \cdot \rho)$, as desired.

Conversely, suppose $x \in f(\theta\cdot\rho)$. In particular, there is some $\tau \subseteq \theta\cdot\rho$ and $n \in \omega$ such that $x \in f_n(\tau)$. There exists some $\sigma \subseteq \theta$ and a $u \in C$ such that $\tau \subseteq \sigma \cdot \psi(u)$; this holds since for any $k \in \omega$, $(\theta \cdot \rho)(k)$ only requires a finite portion of θ and ρ to compute, and τ is finite. It follows that $\langle x, u\rangle \in f(\theta)$ and hence $x \in f(\theta) \cdot f(\rho)$.

This concludes the verification. It remains to show that this embedding restricts to an embedding $\mathcal{B}^X \to \mathcal{G}^{X\oplus\overline{X}}$. Let $\theta \in \mathcal{B}^X$. We show that $f(\theta) \leqslant_e X \oplus \overline{X}$.

We show that for every $n \in \omega$, there exists a computable function g_n such that for every code e of a finite partial X-computable function σ,

$$f_n(\sigma) = \Psi^{X\oplus\overline{X}}_{g_n(e)}.$$

This is sufficient because in order to enumerate $f(\theta)$ for an arbitrary $\theta \in \mathcal{B}$, it suffices to enumerate the codes e of its finite subfunctions and take the union of $g_n(e)$ for each n. This can be done effectively, since given a code d of θ we can effectively find all such e.

We proceed to prove the claim by induction on n. In the base case, we can enumerate the graph of σ given the code using $X \oplus \overline{X}$. For the inductive case of $n+1$, we can enumerate all $u \in C$, all finite partial functions τ, and use g_n to enumerate all $x \in f_n(\tau)$. By enumerating τ in a way that allows us to decide its domain, the relation $\tau \subseteq \sigma \cdot \psi(u)$ can be enumerated. □

By Theorem 7.3 we have that $\mathcal{K}_2 \hookrightarrow \mathcal{B} \hookrightarrow \mathcal{G}$, since the first embedding is just given by the inclusion. At this moment we do not know the status of any of the embeddings in the other direction.

Question 7.4. *Does there exist an embedding $\mathcal{G} \hookrightarrow \mathcal{B}$ or $\mathcal{B} \hookrightarrow \mathcal{K}_2$?*

Theorem 7.3 immediately gives the following result about the c.e. sets $\mathcal{E}$.

Corollary 7.5. *There is a sequence of embeddings $\mathcal{K}_1 \hookrightarrow \mathcal{K}_2^{\text{eff}} \hookrightarrow \mathcal{B}^{\text{eff}} \hookrightarrow \mathcal{E}$.*

Proof. The first embedding exists by Theorem 1.2, the second is by inclusion, and the third is immediate from Theorem 7.3 since $\mathcal{G}^{\emptyset\oplus\overline{\emptyset}} = \mathcal{G}^{\emptyset} = \mathcal{E}$. □

In fact, the embedding given by Theorem 7.3 is tight in the following sense.

Theorem 7.6. *Let $X, Y \subseteq \omega$. There exists an embedding $\mathcal{B}^X \hookrightarrow \mathcal{G}^Y$ if and only if $X \oplus \overline{X} \leqslant_e Y$.*

Proof. One direction follows from Theorem 7.3 and the fact that $G^{X\oplus\overline{X}} \subseteq G^Y$ when $X \oplus \overline{X} \leqslant_e Y$.

For the other direction we turn to Theorem 6.10. Suppose that $f : \mathcal{B}^X \hookrightarrow \mathcal{G}^Y$ is an embedding. By Lemma 6.11, there exist $\theta_\perp$ and $\theta_\top$ such that $f(\theta_\perp)$ and $f(\theta_\top)$ are not comparable with respect to the subset order on $\mathcal{G}^Y$. Let d be a Y-computable $(f(\theta_\perp), f(\theta_\top))$-decider for $X \oplus \overline{X}$.

Pick $i \in f(\theta_\perp) - f(\theta_\top)$ and $j \in f(\theta_\top) - f(\theta_\perp)$. We can enumerate $X \oplus \overline{X}$ now as follows: for each n, we enumerate $\Psi^Y_{d(n)}$ until we see i or j. Note that $\Psi^Y_{d(n)}$ is by construction one of $f(\theta_\perp)$ or $f(\theta_\top)$. If we see i, then $\Psi^Y_{d(n)} \neq f(\theta_\top)$ and thus $n \notin X$, so we enumerate $2i + 1$ into $X \oplus \overline{X}$. If we see j, the converse is the case and we enumerate $2i$ instead.

We hence see $X \oplus \overline{X} \leqslant_e Y$, as desired. □

We would like to prove an analogue of Theorem 6.6 for Scott's graph model, with Turing reducibility replaced by e-reducibility. However, there is a difficulty: the operations we can perform in $\mathcal{G}$ are monotone, while an embedding $f : \mathcal{G}^X \to \mathcal{G}^Y$ need not be so.

Theorem 7.7. *Let $X, Y \subseteq \omega$, $\mathcal{A}$ a pca, and $\gamma : \omega \to \mathcal{A}$ a Y-effectively pca-valued partial numbering. Given an embedding $f : \mathcal{G}^X \hookrightarrow \mathcal{A}$ and any $B_\perp \subseteq B_\top \in \mathcal{G}^X$, there is a Y-computable $(f(B_\perp), f(B_\top))$-decider d for any $Z \leqslant_e X$.*

Proof. Note that $Z \in \mathcal{G}^X$. As before, we construct elements h and t and use the expression $h(\overline{n}tZ)$ to denote either $B_\perp$ or $B_\top$, depending on whether $n \in Z$.

Define, for all $A \subseteq \omega$, $n \in \omega$

$$n \in tA \Leftrightarrow n + 1 \in A$$

$$hA = \begin{cases} B_\perp & \text{if } 0 \notin A \\ B_\top & \text{if } 0 \in A \end{cases}.$$

Now let $d(n)$ be a γ-code of $f(h(\overline{n}tZ))$, which can be constructed from the codes of the subexpressions using ψ as usual. □

Note that the condition $B_\perp \subseteq B_\top$ is essential for h to be well-defined, since we cannot use the absence of 0 in A to enumerate elements into hA.

Lemma 7.8. *The mapping $\gamma(n) = \Psi^X_n$ is an effectively pca-valued partial numbering.*

Proof. It suffices to show that application can be simulated on codes using a computable function, which is the case by the S-m-n theorem. □

The relation $\not\sim_\gamma$ is not generally X-c.e.: even for the case $X = \emptyset$, given codes of two c.e. sets we cannot tell whether they code the same set. However, fixing a set A and an $n \notin A$, we can enumerate the codes of all sets B such that $n \in B$. As in the case of the van Oosten model in Theorem 6.12, the key is to find an $(a_\perp, a_\top)$-decider where $a_\perp$ and $a_\top$ are related in a suitable manner. For our purposes it suffices that $a_\top \not\subseteq a_\perp$, hence our interest is in those embeddings that do not reverse the inclusion order.

Definition 7.9. An embedding $f : \mathcal{G}^X \to \mathcal{G}^Y$ is *order-reversing* if for all $A, B \in \mathcal{G}^X$, whenever $A \subseteq B$, $f(B) \subseteq f(A)$.

Theorem 7.10. *Let $X, Y \subseteq \omega$. There exists a non-order-reversing embedding $f : \mathcal{G}^X \to \mathcal{G}^Y$ if and only if $X \leqslant_e Y$.*

Proof. One direction is trivial, since e-reducibility is transitive, and hence if $A \leqslant_e X$ and $X \leqslant_e Y$ then also $A \leqslant_e Y$. It follows that the inclusion is an order-preserving embedding.

For the other direction, suppose that $f : \mathcal{G}^X \to \mathcal{G}^Y$ is a non-order-reversing embedding and fix sets $B_\perp \subsetneq B_\top \in \mathcal{G}^X$ such that $f(B_\top) \not\subseteq f(B_\perp)$. By Theorem 7.7 there is a computable $(f(B_\perp), f(B_\top))$-decider for X. Fix now an element $k \in f(B_\top) - f(B_\perp)$, which exists by our choice of $B_\perp$ and $B_\top$. In order to enumerate X, we enumerate $\Psi^Y_{d(n)}$ looking for k, and enumerate n into X if we find it, which shows that $X \leqslant_e Y$. Note that d can be taken to be computable since the map $n \mapsto \Psi^Y_n$ is effectively pca-valued. □

We conjecture that Theorem 7.10 can be generalized to arbitrary embeddings $f : \mathcal{G}^X \to \mathcal{G}^Y$. For the record, we pose the following question:

Question 7.11. *Do there exist $X, Y \subseteq \omega$ such that $\mathcal{G}^X \hookrightarrow \mathcal{G}^Y$ and $X \not\leqslant_e Y$?*

It is tempting to try to resolve the question by showing that order-reversing embeddings do not exist, but this is unfortunately not the case, as we now show.

Theorem 7.12. *There exists an order-reversing embedding $\mathcal{E} \to \mathcal{G}$.*

Proof. We define f inductively as follows:

$$\begin{aligned} f_0(A) &= \{e^* \mid A \subseteq W_e\} \\ f_{n+1}(A) &= \{\langle x, 2^{e^*}\rangle \mid e \in \omega \wedge x \in f_n(AW_e)\} \\ f(A) &= \bigcup_{n\in\omega} f_n(A). \end{aligned}$$

By induction on n, we can show that every f_n is order-reversing. The base case is apparent. For the inductive case, note that if $A \subseteq A'$ then $AW_e \subseteq A'W_e$ hence by induction, $f_n(A'W_e) \subseteq f_n(AW_e)$.

It remains to show that f is an embedding. Injectivity follows from the definition of $f_0(A)$: we can recover A by

$$A = \bigcap_{e^* \in f_0(A)} W_e$$

and no $f_{n+1}(A)$ contains elements of the form e^*.

To see $f(AB) = f(A)f(B)$, let $x \in f(AB)$; in particular, $x \in f_n(AB)$ for some $n \in \omega$. Take e such that $W_e = B$, then $\langle x, 2^{e^*}\rangle \in f_{n+1}(A) \subset f(A)$. By our choice of e, $e^* \in f_0(B) \subset f(B)$. Hence $x \in f(A)f(B)$, as desired.

For the other direction, let $x \in f(A)f(B)$. Then there is some e such that $\langle x, 2^{e^*}\rangle \in f(A)$ and $e^* \in f(B)$. Thus $x \in f(AW_e)$, and since $e^* \in f(B)$, $B \subseteq W_e$. Since f is order-reversing, $f(AW_e) \subseteq f(AB)$, as desired. □

We can use Theorem 7.7 to clarify the relation between $\mathcal{G}^X$ and $\mathcal{K}_2^Y$ as follows. The reader may compare this to Corollary 6.15.

Theorem 7.13. *Let $X, Y \subseteq \omega$. If there exists an embedding $f : \mathcal{G}^X \hookrightarrow \mathcal{K}_2^Y$ then every $Z \leqslant_e X$ is Y-computable.*

Proof. Let $\gamma_t : \omega \to \mathcal{K}_2^Y$ be the effectively pca-valued partial numbering from Lemma 6.5 and let $B_\bot \subsetneq B_\top \in \mathcal{G}^X$. By Theorem 7.7 there is a computable $(f(B_\bot), f(B_\top))$-decider for Z. Since γ_t has Y-c.e. inequivalence, by Theorem 3.7 we have $Z \leqslant_T Y$. □

Theorem 7.13 has the following two results as corollaries, which show that Corollaries 7.5 and 7.2 cannot be reversed.

Corollary 7.14. *$\mathcal{E}$ does not embed in $\mathcal{K}_2^{\text{eff}}$.*

Proof. Suppose an embedding $\mathcal{E} \hookrightarrow \mathcal{K}_2^{\text{eff}}$ existed. Then by Theorem 7.13 the halting set K would be decidable, a contradiction.

Note that we can also derive the corollary from Corollaries 6.16 and 7.5: By Corollary 6.16 we have that $\mathcal{B}^{\text{eff}} \not\hookrightarrow \mathcal{K}_2^{\text{eff}}$, so if $\mathcal{E} \hookrightarrow \mathcal{K}_2^{\text{eff}}$ then by Corollary 7.5 we obtain $\mathcal{B}^{\text{eff}} \hookrightarrow \mathcal{E} \hookrightarrow \mathcal{K}_2^{\text{eff}}$, contradicting Corollary 6.16. □

Corollary 7.15. *$\mathcal{E}$ does not embed in $\mathcal{K}_1$.*

Proof. Suppose an embedding $\mathcal{E} \hookrightarrow \mathcal{K}_1$ existed. Since $\mathcal{K}_1$ embeds in $\mathcal{K}_2^{\text{eff}}$ (Theorem 1.1), there would be an embedding $\mathcal{E} \hookrightarrow \mathcal{K}_2^{\text{eff}}$, contradicting Corollary 7.14. □

References

[1] A. Asperti and A. Ciabattoni, *A sufficient condition for completability of partial combinatory algebras*, Journal of Symbolic Logic 64(4) (1997) 1209–1214.

[2] H. P. Barendregt, *The lambda calculus*, Studies in Logic and the Foundations of Mathematics Vol. 103, North-Holland, Amsterdam, 1984 (2nd edition).

[3] M. J. Beeson, *Foundations of constructive mathematics*, Springer-Verlag, 1985.

[4] I. Bethke, *Notes on Partial Combinatory Algebras*, PhD thesis, Universiteit van Amsterdam, 1988.

[5] I. Bethke, J. W. Klop, and R. de Vrijer, *Completing partial combinatory algebras with unique head-normal forms*, technical report CS-R9544, Centrum Wiskunde & Informatica (CWI), 1995.

[6] Y. L. Ershov, *Theorie der Numerierungen I*, Zeitschrift für mathematische Logik und Grundlagen der Mathematik 19 (1973) 289–388.

[7] S. Feferman, *A language and axioms for explicit mathematics*, in: J. N. Crossley (ed.), Algebra and Logic, Springer, 1975, 87–139.

[8] H. Friedman, *Axiomatic recursive function theory*, in: Gandy and Yates [10], 113–137.

[9] E. Frittaion and M. Rathjen, *Extensional realizability for intuitionistic set theory*, Journal of Logic and Computation 31(2) (2021) 630–653.

[10] R. O. Gandy and C. E. M. Yates (eds.), *Logic Colloquium '69*, Proceedings of the Summer School and Colloquium in Mathematical Logic, Manchester, 1969, Amsterdam: North-Holland, 1971.

[11] S. C. Kleene and R. E. Vesley, *The foundations of intuitionistic mathematics*, North-Holland, 1965.

[12] J. Longley and D. Normann, *Higher-order computability*, Springer, 2015.

[13] J. Longley *Realizability Toposes and Language Semantics*, PhD thesis, University of Edinburgh, 1994.

[14] D. C. McCarty, *Realizability and recursive set theory*, Annals of Pure and Applied Logic 32(2) (1986) 153–183.

[15] Y. N. Moschovakis, *Axioms for computation theories—first draft*, in: Gandy and Yates [10], 199–255.

[16] P. Odifreddi, *Classical recursion theory*, Vol. 1, Studies in logic and the foundations of mathematics Vol. 125, North-Holland, Amsterdam, 1989.

[17] P. G. Odifreddi, *Classical recursion theory*, Vol. 2, Studies in logic and the foundations of mathematics Vol. 143, North-Holland, Amsterdam, 1999.

[18] J. van Oosten, *Realizability: An introduction to its categorical side*, Studies in logic and the foundations of mathematics Vol. 152, Elsevier, 2008.

[19] J. van Oosten, *A combinatory algebra for sequential functionals of finite type*, S. B. Cooper and J. K. Truss (eds.), Models and Computability, Cambridge University Press (1999) 389–406.

[20] M. Rathjen, *Realizability for constructive Zermelo-Fraenkel set theory*, in: J. Väänänen, V. Stoltenberg-Hansen (eds.), Proceedings of the Logic Colloquium 2003, Lecture Notes In Logic 141 (2006) 442–471.

[21] D. Scott, *Lambda calculus and recursion theory (preliminary version)*, in: S. Kanger (ed.), Proceedings of the Third Scandinavian Logic Symposium, Studies in Logic and the Foundations of Mathematics Vol. 82 (1975) 154–193.

[22] P. Shafer and S. A. Terwijn, *Ordinal analysis of partial combinatory algebras*, Journal of Symbolic Logic 86(3) (2021) 1154–1188.

[23] R. I. Soare, *Recursively enumerable sets and degrees*, Springer-Verlag, 1987.

[24] H. R. Strong, *Algebraically generalized recursive function theory*, IBM Journal of Research and Development 12 (1968) 465–475.

[25] A. S. Troelstra and D. van Dalen, *Constructivism in Mathematics*, Vol. 2, Studies in logic and the foundations of mathematics Vol. 123, North-Holland, 1988.

[26] E. G. Wagner, *Uniform reflexive structures: on the nature of Gödelizations and relative computability*, Transactions of the AMS 144 (1969) 1–41.

Appendix A. Equivalence of codings

In section 6, we introduced alternative codings for Kleene's second model $\mathcal{K}_2$ and for van Oosten's sequential computation model $\mathcal{B}$. Let us now show that these can indeed be considered to be merely different codings of the same models, and that our results about embeddings extend to the usual codings. Note that since our results are often about relativizations of $\mathcal{K}_2$ and $\mathcal{B}$, it is essential that these embeddings also restrict to embeddings between the relativized models as well.

We will use $\mathcal{K}_2$ and $\mathcal{B}$ to refer to the usual codings, as they are presented in [18], and $\mathcal{P}$ and $\mathcal{P}_\perp$ respectively to refer to our codings. Our encodings use the same carrier sets, the set ω^ω of total functions $\omega \to \omega$ and the set $\omega^\omega_\perp$ of partial functions $\omega \rightharpoonup \omega$, respectively. For completeness, and to introduce a number of useful shorthands, we will now restate the definitions.

Recall that we have a bijective coding scheme $\langle \dots \rangle$ on the naturals. We will use $\vec{\alpha}$ to refer to a finite sequence of naturals, and $\langle \vec{\alpha} \rangle$ or just α to refer to its code. We use $|\alpha|$ to mean the number of elements of α, and α_i to retrieve the i-th element of α (starting from 0), seen as the code of a sequence. Given $n, \vec{\alpha} \in \omega$, let $n \triangleright \langle \vec{\alpha} \rangle = \langle n, \vec{\alpha} \rangle$, $\langle \vec{\alpha} \rangle \triangleleft n = \langle \vec{\alpha}, n \rangle$, and let $\mathrm{ask}(n) = 2n$ and $\mathrm{say}(n) = 2n + 1$. Given $x, \xi \in \omega$ we will use $x \triangleright^* \xi$

to denote $((x \rhd \xi_0) \rhd \ldots) \rhd \xi_{|\xi|-1}$; note that here each ξ_i is itself interpreted as a sequence.[5] Given a function $f : \omega \to \omega$ and $n \in \omega$, let $f \restriction n$ be $\langle f(0), \ldots, f(n-1) \rangle$. Given $d, e \in \omega$, define $d * e$ as the following oracle computation for an arbitrary oracle ψ:

$$\Phi^{\psi}_{d*e} = \Phi^{\Phi^{\psi}_{e}}_{d}. \tag{5}$$

That is, to execute $d * e$, simulate d and use e to compute any oracle queries that d performs, using the oracle ψ for any oracle queries of e. Note that $*$ is associative, and has an identity element e_* satisfying $\Phi^{\psi}_{e_*} = \psi$.

For the usual coding of Kleene's second model $\mathcal{K}_2$, it is helpful to introduce the following notion of a limit. Given a sequence of naturals a_n and $k \in \omega$, we say *a converges* to k if there exists an n such that $a_n = k + 1$ and for all $i < n$, $a_i = 0$. In this case, we write $\lim_{n\to\infty} a_n = k$. Note that n here does not truly "go to infinity", since we only look up to the first non-zero element; however, we trust this will not cause confusion.

We extend this notion pointwise to families of functions, where given a family $f_n : \omega \to \omega$ and a function $g : \omega \to \omega$, we say $\lim_{n\to\infty} f_n = g$ when for every $x \in \omega$, $\lim_{n\to\infty} f_n(x) = g(x)$.

Given $f, g \in \mathcal{K}_2$, their composition in $\mathcal{K}_2$ exists and is given by

$$f \cdot g = \lim_{n\to\infty} \lambda x.\, f(x \rhd (g \restriction n))$$

if the right hand side exists (in which case it is necessarily unique). If the limit does not exist, then $f \cdot g$ is undefined.

The usual coding of van Oosten's sequential computations $\mathcal{B}$ is defined in terms of *dialogues*, which we will define in terms of a limit-like construction similar to the one above.

Given partial functions $\varphi, \psi : \omega \rightharpoonup \omega$ and $k \in \omega$, we say that φ ψ-converges to k if there exists an α such that $\varphi(\alpha) = \mathrm{say}(k)$ and for every proper initial segment β of α, there exists some l such that $\varphi(\beta) = \mathrm{ask}(l)$ and $\alpha_{|\beta|} = \psi(l)$. We write $\lim_{\alpha\to\psi} \varphi(\alpha) = k$ when this is the case.

We again extend this notion pointwise to functions; given δ_α a family of functions, we write $\lim_{\alpha\to\psi} \delta_\alpha = \Delta$ for some Δ iff for every x, $\lim_{\alpha\to\psi} \delta_\alpha(x) \simeq \Delta(x)$.

Given two partial functions $\varphi, \psi \in \mathcal{B}$, their composition in $\mathcal{B}$ is defined by

$$(\varphi \cdot \psi)(x) = \lim_{\alpha\to\psi} \varphi(x \rhd \alpha).$$

The application $\varphi \cdot \psi$ is thus always defined.

Note that if α is the sequence witnessing that the limit exists for x, the sequence $x \rhd \alpha$ is what van Oosten [18] calls a *dialogue* between φ and ψ.

Note that these definitions suffice to give us the relativized versions of $\mathcal{K}_2$ and $\mathcal{B}$ simply by restricting the carrier set.

We are now ready to show that our codings $\mathcal{P}$ and $\mathcal{P}_\perp$ are mutually embeddable with $\mathcal{K}_2$ and $\mathcal{B}$ respectively. Given $e \in \omega$ and φ a partial

[5] This operation may resemble concatenation but is in fact *not* that: the result has length $|\xi_{|\xi|-1}| + 1$, and for all $x, \xi, \alpha \in \omega$, $(x \rhd^* \xi) \rhd \alpha = x \rhd^* (\xi \lhd \alpha)$.

function, define

$$(S_e\varphi)(0) = e$$
$$(S_e\varphi)(n+1) = \varphi(n).$$

We understand $(S_e\varphi)(n+1)$ to be undefined whenever $\varphi(n)$ is undefined. The embeddings of $\mathcal{K}_2$ and $\mathcal{B}$ into our codings will be given by S_e for particular choices of e. The embeddings in the other direction are rather more involved.

Theorem A.1. *There exists an $e \in \omega$ such that S_e is an embedding $\mathcal{K}_2 \to \mathcal{P}$ and S_e restricts to an embedding $\mathcal{K}_2^X \to \mathcal{P}^X$ for every oracle X.*

Proof. We define e as the least natural such that for all $f, g \in \omega^\omega$ and $x, d \in \omega$,

$$\Phi_e^{S_d f \oplus S_d g}(0) = d$$
$$\Phi_e^{S_d f \oplus S_d g}(x+1) = \lim_{n\to\infty} f(x \triangleright (g \restriction n)).$$

It is then easy to verify that if $f \cdot g\downarrow$ then

$$\begin{aligned}(S_e f \cdot S_e g)(0) &= e\\ (S_e f \cdot S_e g)(x+1) &= \Phi_e^{S_e f \oplus S_e g}(x)\\ &= \lim_{n\to\infty} f(x \triangleright (g \restriction n))\\ &= (f \cdot g)(x) = S_e(f \cdot g)(x+1),\end{aligned}$$

as desired.

Since $S_e f$ is f-computable, the embedding restricts to an embedding for every relativization. □

Theorem A.2. *There exists an $e \in \omega$ such that S_e is an embedding $\mathcal{B} \to \mathcal{P}_\perp$ and S_e restricts to an embedding $\mathcal{B}^X \to \mathcal{P}_\perp^X$ for every oracle X.*

Proof. The proof is analogous to the proof of Theorem A.1, with the difference that e simulates the application in $\mathcal{P}_\perp$ instead of in $\mathcal{P}$. □

The constructions in the other direction are considerably more involved, as the embedding of a (partial) function ψ needs to encode the behavior of the code $\psi(0)$. For convenience, we fix a code $r \in \omega$ such that for every oracle ψ,

$$\Phi_r^\psi(x) = \Phi_{\psi(0)}^\psi(x) \tag{6}$$

which will simplify the following construction. Note that for $f, g \in \mathcal{P}$, we have that $f \cdot g = \Phi_r^{f \oplus g}$ iff $f \cdot g\downarrow$, and similarly for $\mathcal{P}_\perp$ (where the termination condition holds trivially).

In order to simplify our arguments, let us also introduce some further terminology. When talking about a (partial) function $\varphi : \omega \rightharpoonup \omega$, we will use *places* to refer to elements of its domain. We will say that a place x is *coding* for φ if it can play a role when φ is applied to another element, and *non-coding* if it cannot. Concretely, for $f \in \mathcal{K}_2$ a place $x \triangleright \alpha$ is coding if for all proper initial segments β of α, $f(x \triangleright \beta) = 0$, and it is non-coding otherwise. For $\varphi \in \mathcal{B}$, a place $x \triangleright \alpha$ is coding if for all proper initial segments β of α, $\varphi(x \triangleright \beta)$ is of the form ask(k) for some k.

Theorem A.3. *There exists an embedding $E : \mathcal{P} \to \mathcal{K}_2$, which restricts to an embedding $\mathcal{P}^X \to \mathcal{K}_2^X$ for every oracle X.*

Proof. The proof proceeds in two steps: we first define $E : \mathcal{P} \to \mathcal{K}_2$, and then show that it is in fact an embedding; that is, that for all $f, g \in \mathcal{P}$, if $f \cdot g\downarrow$ then $Ef \cdot Eg\downarrow = E(f \cdot g)$.

For the construction of E, we will separate ω into two sets, X and Y. For every $f \in \mathcal{P}$, we will use the places in X to encode f in Ef and ensure that this encoding is preserved under application. The places in Y will be given the same values in Ef regardless of the choice of f.

Let ζ be the all-zero sequence of length $\langle\rangle + 1$. For $x \in \omega$ let $\dot{x} = \langle\rangle \triangleright \zeta \triangleleft x$. We will ensure that for every $f \in \mathcal{P}$, $\dot{x}$ is non-coding for Ef, so that we can set $(Ef)(\dot{x}) = f(x)$ and thereby encode f in Ef. Given a sequence α, let $\hat{\alpha}$ be the shortest sequence such that for all k with $\dot{k} < |\alpha|$, $\hat{\alpha}_k = \alpha_{\dot{k}}$.

Let

$$X = \{\dot{x} \triangleright^* \xi \mid x, \xi \in \omega\}$$
$$Y = \omega - X.$$

For $n, \alpha \in \omega$, let $r_{n,\alpha}$ be a code for the following computation with an arbitrary total oracle f. On input x, simulate $\Phi_r^{f \oplus \hat{\alpha}}(x)$ for $|\alpha|$ steps, where r is the code defined in (6). When f is queried and returns k, if $k < n$ then terminate the whole computation and return k. If $k \geqslant n$, then use $k - n$ as the result of the oracle query. If $\hat{\alpha}$ is queried return the value at that position or, if it is out of bounds, terminate the whole computation and return n. Similarly, if the computation runs out of steps, return n. Finally, if the computation terminates with result k, return $k + n + 1$.

The definition of $r_{n,\alpha}$ can be understood as follows: $r_{n,\alpha}$ is a computation with a left oracle that may fail in n different ways (denoted by responses 0 through $n - 1$), and a right oracle that is partially known ($\hat{\alpha}$). Hence, $r_{n,\alpha}$ is a computation which may fail in $n + 1$ different ways.

Now define, using $*$ as defined in (5),

$$L(\langle\rangle) = e_*$$
$$L(\alpha \triangleright \xi) = r_{|\xi|,\alpha} * L(\xi).$$

Given some possibly undefined expression e, let $\mathrm{try}(e)$ be $n + 1$ if e has value n, and 0 otherwise.

Using this, we can now define Ef:

$$\begin{aligned} (Ef)(\dot{x} \triangleright^* \xi) &= \Phi^f_{L(\xi)}(x) && \text{for } x, \xi \in \omega \\ (Ef)(\langle\rangle) &= 0 \\ (Ef)(y \triangleright \alpha) &= \mathrm{try}(\alpha_y) && \text{for } y \triangleright \alpha \in Y. \end{aligned}$$

The last line of this definition can be read as $(Ef)(y \triangleright \alpha) = \alpha_y + 1$ when $y < |\alpha|$ and $(Ef)(y \triangleright \alpha) = 0$ otherwise.

We immediately see that Ef is in fact f-computable, from which it follows that if E is an embedding $\mathcal{P} \to \mathcal{K}_2$, then it restricts to an embedding $\mathcal{P}^X \to \mathcal{K}_2^X$ for every oracle X. Moreover, for any $x \in \omega$,

$$(Ef)(\dot{x}) = \Phi^f_{L(\langle\rangle)}(x) = \Phi^f_{e_*}(x) = f(x),$$

meaning that E is injective.

It thus remains to show that for all $f, g \in \mathcal{P}$, if $f \cdot g{\downarrow}$ then $Ef \cdot Eg{\downarrow} = E(f \cdot g)$, which we will do pointwise. We will first show this for Y, since it is simpler, and then for X.

Let $f, g \in \mathcal{P}$, $y \in Y$, and suppose $f \cdot g{\downarrow}$. Note that for all $k \in \omega$,

$$\lim_{n\to\infty} \operatorname{try}((Eg \upharpoonright n)_k) = (Eg)(k),$$

since eventually $n > k$.

We consider two cases for y. If $y = \langle\rangle$, we have

$$\lim_{n\to\infty} (Ef)(\langle\rangle \triangleright (Eg \upharpoonright n)) = \lim_{n\to\infty} \operatorname{try}((Eg \upharpoonright n)_{\langle\rangle}) = (Eg)(\langle\rangle) = 0 = E(f \cdot g)(\langle\rangle).$$

If $y = y' \triangleright \alpha$, we have

$$\begin{aligned}\lim_{n\to\infty} (Ef)((y' \triangleright \alpha) \triangleright (Eg \upharpoonright n)) &= \lim_{n\to\infty} \operatorname{try}((Eg \upharpoonright n)_{y' \triangleright \alpha}) \\ &= (Eg)(y' \triangleright \alpha) = \operatorname{try}(\alpha_{y'}) = E(f \cdot g)(y' \triangleright \alpha).\end{aligned}$$

It follows that for all $y \in Y$,

$$\lim_{n\to\infty} (Ef)(y \triangleright (Eg \upharpoonright n)) = E(f \cdot g)(y),$$

as desired.

Now let us consider the case for X, which is more complicated. Let $f, g \in \mathcal{P}$, $\dot{x} \triangleright^* \xi \in X$, and assume $f \cdot g{\downarrow}$.

We wish to show

$$\begin{aligned}(Ef \cdot Eg)(\dot{x} \triangleright^* \xi) &= \lim_{n\to\infty} (Ef)((\dot{x} \triangleright^* \xi) \triangleright (Eg \upharpoonright n)) \\ &= \lim_{n\to\infty} (Ef)(\dot{x} \triangleright^* (\xi \triangleleft (Eg \upharpoonright n))) \\ &= \lim_{n\to\infty} \Phi^f_{L(\xi \triangleleft (Eg \upharpoonright n))}(x) \\ &\overset{!}{=} \Phi^{f \cdot g}_{L(\xi)}(x) \\ &= E(f \cdot g)(\dot{x} \triangleright^* \xi),\end{aligned}$$

where all except the marked equality are easily verified. We will prove the marked equality by induction on the length of ξ.

For the base case, note that

$$\lim_{n\to\infty} \Phi^f_{L(\langle\rangle \triangleleft (Eg \upharpoonright n))}(x) = \lim_{n\to\infty} \Phi^f_{L(\langle Eg \upharpoonright n\rangle)}(x) = \lim_{n\to\infty} \Phi^f_{r_{0,Eg \upharpoonright n}}(x)$$

and

$$\Phi^{f \cdot g}_{L(\langle\rangle)}(x) = \Phi^{f \cdot g}_{e_*}(x) = (f \cdot g)(x) = \Phi^{f \oplus g}_r(x).$$

We will refer to f as the left oracle and g or $g \upharpoonright n$ as the right oracle, and refer to $\Phi^f_{r_{0,Eg \upharpoonright n}}(x)$ and $\Phi^{f \oplus g}_r(x)$ as the former and latter computation respectively. The former and latter computations will behave identically except if the former performs an unsuccessful oracle query (which is necessarily to the right oracle, since the left oracle is total) or runs out of steps, in which case it will terminate with result 0. If this does not occur and both computations terminate, they will, for some k, terminate with result $k + 1$ and k respectively, and hence satisfy the conditions for the latter to be the limit of the former, provided the computations do indeed terminate. This is the case, since the latter computation only performs finitely many queries to the oracle, and so for a sufficiently large n the former computation will have no unsuccessful oracle queries, and the two will both run to completion.

For the inductive step, suppose that for $\xi \in \omega$ we have already shown

$$\lim_{n\to\infty} \Phi^{f}_{L(\xi \lhd (Eg\restriction n))} = \Phi^{f\cdot g}_{L(\xi)}.$$

We now wish to show that for all $\alpha \in \omega$,

$$\lim_{n\to\infty} \Phi^{f}_{L(\alpha \rhd \xi \lhd (Eg\restriction n))} = \lim_{n\to\infty} \Phi_{r_{|\xi|+1,\alpha}}^{\Phi^{f}_{L(\xi \lhd (Eg\restriction n))}} \overset{!}{=} \Phi_{r_{|\xi|,\alpha}}^{\Phi^{f\cdot g}_{L(\xi)}} = \Phi^{f\cdot g}_{L(\alpha \rhd \xi)},$$

where again only the innermost equality is of interest; the rest follow by expanding the definitions.

For convenience, we will write

$$h_n = \Phi^{f}_{L(\xi \lhd (Eg\restriction n))} \text{ and } H = \Phi^{f\cdot g}_{L(\xi)}.$$

Our induction hypothesis is thus $\lim_{n\to\infty} h_n = H$, and our goal, letting $m = |\xi|$, is to show that for all $x, \alpha \in \omega$,

$$\lim_{n\to\infty} \Phi^{h_n}_{r_{m+1,\alpha}}(x) = \Phi^{H}_{r_{m,\alpha}}(x).$$

As in the base case, for any x, the two computations behave identically except for oracle queries and step bounds, and except that if both run to completion, the left will return a value one greater than the right, which is exactly what we want for the limit. We again refer to h_n or H as the left oracle and α as the right oracle, and to $\Phi^{h_n}_{r_{m+1,\alpha}}(x)$ and $\Phi^{H}_{r_{m,\alpha}}(x)$ as the former and latter computations respectively.

Let us consider the possible oracle queries:

- If the computations query the right oracle, it is either out of bounds, in which case both computations will terminate and will return $m+1$ and m respectively, or it is in bounds, in which case both will see the same oracle response. The same occurs if the computations hit the step bounds.
- If the computations query the left oracle, and the former computation receives 0, then it will immediately terminate with result 0.
- If the computations query the left oracle, and the former computation receives $k+1 < m+1$, then the latter computation will receive $k < m$, and both will terminate, with result $k+1$ and k respectively.
- If the computations query the left oracle, and the former computation receives $k+1 \geq m+1$, then the latter computation will receive $k \geq m$, and both use $k+1-m+1 = k-m$ as the oracle response.

We thus see that unless the former computation terminates with result 0 (which does not affect the limit behavior), the two computations will indeed return corresponding results, with the result of the former computation one greater, as required by the definition of a limit. Again, it remains to show that the former computation will eventually give a non-zero response, and this is indeed the case: namely, as before, the latter computation will query the left oracle only finitely often, and there exists some $n \in \omega$ such that h_n gives a non-zero result for those finitely many queries.

It follows that the limit indeed holds, which concludes the inductive case. We thus have that $(Ef \cdot Eg)(\dot{x} \rhd^* \xi) = E(f \cdot g)(\dot{x} \rhd^* \xi)$, as desired, and hence E is indeed an embedding. □

Theorem A.4. *There exists an embedding $E : \mathcal{P}_\perp \to \mathcal{B}$, which restricts to an embedding $\mathcal{P}_\perp^X \to \mathcal{B}^X$ for every oracle X.*

Proof. The structure of this proof is similar to that of Theorem A.3, but the details of the encoding differ considerably. Once again, we will first construct $E : \mathcal{P}_\perp \to \mathcal{B}$ and then show that it is an embedding. Since application in $\mathcal{P}_\perp$ and $\mathcal{B}$ is total, it suffices to show that for all $\varphi, \psi \in \mathcal{P}_\perp$, $E\varphi \cdot E\psi = E(\varphi \cdot \psi)$.

Given $x \in \omega$, let $\dot{x} = \langle \langle \rangle, 0, x \rangle$. We split ω into X and Y as follows:

$$\begin{aligned} X &= \{ \dot{x} \rhd^* \xi \mid x, \xi \in \omega \} \\ Y &= \omega - X, \end{aligned}$$

In the proof of Theorem A.3, we used the fact that if a sequence of functions f_n had a limit g, then for all i and x, $f_i(x)$ was an approximation of $g(x)$. This is not the case here: given a family of functions δ_α with limit Δ, for every x we will in general need a different β for $\delta_\beta(x)$ to be an approximation of $\Delta(x)$, since β should contain responses to the queries made by $\delta_{\beta'}(x)$ for $\beta' \sqsubset \beta$. To work around this problem, we thread the responses provided to us in ξ through our computation, allowing subcomputations to consume some of the responses and then return the remaining ones.

Given $n \in \omega$, let $p_{n,m}$ be the function that maps $\mathrm{say}^n(k \rhd^* \xi)$ to $\mathrm{say}^n(k)$ when $|\xi| = m$, and that is the identity everywhere else. Let r_n be defined inductively as follows. Let $r_0 = e_*$, the identity of $*$ as defined in (5). Let r_{n+1} be the code of the function that, on input $(x \rhd \beta) \rhd^* \xi$ where $|\xi| = n$, simulates r on input x, but modifies the oracle queries as follows. On an oracle query of the form $2l$, run r_n on input $l \rhd^* \xi$. If the result is of the form $\mathrm{say}^n(k \rhd^* \xi')$, update ξ to be ξ' and use k as the result of the query. If the result is of any other form k, terminate the computation and return k. On an oracle query of the form $2l + 1$, use the first value in β, removing it from β. If no value is available, return $\mathrm{say}^n(\mathrm{ask}(\dot{l}))$. If r runs to completion with result k, return $\mathrm{say}^{n+1}((k \rhd \beta) \rhd^* \xi)$, where β and ξ have possibly been updated in the process of simulating r.

We define E as follows:

$$\begin{aligned} (E\varphi)(\dot{x} \rhd^* \xi) &= p_{|\xi|,|\xi|}\left(\Phi^\varphi_{r_{|\xi|}}(x \rhd^* \xi)\right) && \text{for } x, \xi \in \omega \\ (E\varphi)(\langle y \rangle) &= \mathrm{ask}(y) && \text{for } y \in Y \\ (E\varphi)(\langle y, z \rangle) &= \mathrm{say}(z) && \text{for } y \in Y, z \in \omega \\ (E\varphi)(y) &= 0 && \text{otherwise.} \end{aligned}$$

Once again, we see that $E\varphi$ is partial φ-computable and that E is injective, as desired. As before, $(E\varphi)(\dot{x}) = \Phi^\varphi_{e_*}(x) = \varphi(x)$ whenever the right-hand side is defined.

To show that E is an embedding, it remains to verify that it preserves application. Let $\varphi, \psi \in \mathcal{P}_\perp$.

For $y \in Y$, we see that

$$(E\varphi \cdot E\psi)(y) = \lim_{\alpha \to E\psi} (E\varphi)(y \rhd \alpha) = (E\psi)(y),$$

where the first equality is by definition and the second is witnessed by the dialogue $\langle y, (E\psi)(y) \rangle$. Since $y \in Y$, we have $(E\psi)(y) = E(\varphi \cdot \psi)(y)$ by our definition of E, as desired. Note that we have no problems with non-termination here; all terms are defined.

Let us now look at X. Our aim is to show that for all $x, \xi \in \omega$, letting $n = |\xi|$, the following sequence of congruences holds:

$$\begin{aligned}(E\varphi \cdot E\psi)(\dot{x} \rhd^* \xi) &\simeq \lim_{\alpha \to E\psi} p_{n+1,n+1}\left(\Phi^{\varphi}_{r_{n+1}}((x \rhd^* \xi) \rhd \alpha)\right)\\ &\simeq p_{n,n+1}\left(\lim_{\alpha \to E\psi} \Phi^{\varphi}_{r_{n+1}}((x \rhd^* \xi) \rhd \alpha)\right)\\ &\overset{!}{\simeq} p_{n,n}\left(\Phi^{\varphi\cdot\psi}_{r_n}(x \rhd^* \xi)\right)\\ &\simeq E(\varphi \cdot \psi)(\dot{x} \rhd^* \xi).\end{aligned}$$

Only the marked equivalence is of interest, as the first and last hold by definition, and the second is easily checked. We show that this equivalence holds by an induction on the length of ξ. Again letting $n = |\xi|$, we show:

- If $\Phi^{\varphi\cdot\psi}_{r_n}(x \rhd^* \xi)$ terminates and is of the form $\mathrm{say}^n(k \rhd^* \xi')$ then
$$\lim_{\alpha \to E\psi} \Phi^{\varphi}_{r_{n+1}}((x \rhd^* \xi) \rhd \alpha) = \mathrm{say}^n((k \rhd^* \xi') \rhd \langle\rangle),$$
and if α is a supersequence of the sequence witnessing the limit, then
$$\Phi^{\varphi}_{r_{n+1}}((x \rhd^* \xi) \rhd \alpha) = \mathrm{say}^{n+1}((k \rhd^* \xi') \rhd \alpha'), \tag{7}$$
where α' is the sequence of oracle responses not consumed by the computation.
- If $\Phi^{\varphi\cdot\psi}_{r_n}(x \rhd^* \xi)$ terminates with any result k not of the above form, then
$$\lim_{\alpha \to E\psi} \Phi^{\varphi}_{r_{n+1}}((x \rhd^* \xi) \rhd \alpha) = k.$$
- If $\Phi^{\varphi\cdot\psi}_{r_n}(x \rhd^* \xi)$ does not terminate, then
$$\lim_{\alpha \to E\psi} \Phi^{\varphi}_{r_{n+1}}((x \rhd^* \xi) \rhd \alpha)$$
does not exist.

To see this suffices, note that if $\Phi^{\varphi\cdot\psi}_{r_n}(x \rhd^* \xi)$ terminates then, whatever the result, we have

$$p_{n,n+1}\left(\lim_{\alpha \to E\psi} \Phi^{\varphi}_{r_{n+1}}((x \rhd^* \xi) \rhd \alpha)\right) = p_{n,n}\left(\Phi^{\varphi\cdot\psi}_{r_n}(x \rhd^* \xi)\right).$$

Conversely, if $\Phi^{\varphi\cdot\psi}_{r_n}(x \rhd^* \xi)$ diverges then both sides diverge, as desired.

Let us now proceed with the inductive proof. For the base case, note that

$$\Phi^{\varphi\cdot\psi}_{r_0}(x) = \Phi^{\varphi\cdot\psi}_{e_*}(x) = (\varphi \cdot \psi)(x) = \Phi^{\varphi\oplus\psi}_{r}(x)$$

and consider the computation $\Phi^{\varphi}_{r_1}(x \rhd \alpha)$ in the context of the limit

$$\lim_{\alpha \to E\psi} \Phi^{\varphi}_{r_1}(x \rhd \alpha). \tag{8}$$

This computation simulates r, and this simulation can do the following:

- Query the oracle at $2l$: it will receive $\varphi(l)$ as response, since $r_0 = e_*$. Note that the response is always of the form $k = \mathrm{say}^0(k \rhd^* \langle\rangle)$.

- Query the oracle at $2l+1$, with no more elements of α available: the computation terminates with response $\mathrm{ask}(\dot{l})$, ensuring α will be extended with $(E\psi)(\dot{l}) = \psi(l)$.
- Query the oracle at $2l+1$, with elements of α remaining: the next element of α is $\psi(l)$, as ensured by the previous point.
- Terminate with result k: the computation as a whole terminates with $\mathrm{say}(k \rhd \alpha)$, where α may have been updated in the process of the computation.

We thus see that if α is sufficiently long and $\Phi_r^{\varphi\oplus\psi}(x)$ terminates, then

$$\Phi_{r_1}^{\varphi}(x \rhd \alpha) = \mathrm{say}\left(\Phi_r^{\varphi\oplus\psi}(x) \rhd \alpha'\right),$$

where α' is the sequence of left-over elements of α after all oracle queries have been handled by the simulation. It follows that

$$\lim_{\alpha\to E\psi} \Phi_{r_{n+1}}^{\varphi}(x \rhd \alpha) = \Phi_r^{\varphi\oplus\psi}(x) \rhd \langle\rangle$$

and (7) holds, as required. If $\Phi_r^{\varphi\oplus\psi}(x)$ diverges, then the limit does not exist, satisfying the last condition of the induction.

This concludes the base case. The argument for the inductive case is analogous, with the computations of interest being

$$\Phi_{r_{n+2}}^{\varphi}(((x \rhd \beta) \rhd^* \xi) \rhd \alpha) \text{ and } \Phi_{r_{n+1}}^{\varphi\cdot\psi}(((x \rhd \beta) \rhd^* \xi),$$

which we will refer to as the former and latter computation respectively.

Both of these computations simulate r, and differ only in their behavior with respect to oracle queries, as well as the result upon termination. While there are many cases, the argument in each case is straightforward. The possible things the simulations can do are:

- Query the oracle at $2l$, with results of the form $\mathrm{say}^{n+1}((k \rhd^* \xi) \rhd \alpha)$ and $\mathrm{say}^{n}(k \rhd^* \xi)$ respectively: both simulations proceed with result k
- Query the oracle at $2l$, with results of the form $\mathrm{say}(k)$ and k respectively: both computations terminate, with result $\mathrm{say}(k)$ and k respectively.
- Query the oracle at $2l$, with a result of the form $\mathrm{ask}(k)$ for the former computation: the former computation terminates with result $\mathrm{ask}(k)$, ensuring α is extended.
- Query the oracle at $2l+1$, and β has elements remaining: both simulations use the next element of β as the value.
- Query the oracle at $2l+1$, and β is empty: the computations terminate with results $\mathrm{say}^{n+1}(\mathrm{ask}(\dot{l}))$ and $\mathrm{say}^{n}(\mathrm{ask}(\dot{l}))$ respectively.
- Both simulations terminate with result k: the computations terminate with result $\mathrm{say}^{n+2}(((k \rhd \beta) \rhd^* \xi) \rhd \alpha)$ and $\mathrm{say}^{n+1}((k \rhd \beta) \rhd^* \xi)$ respectively.

We see that the simulations will proceed in lockstep, and that the relation between the final results will satisfy the definition of a limit, as desired. The conditions of the induction can be checked the same way as in the base case.

Putting it all together, we see that for all $x \in \omega$, $(E\varphi \cdot E\psi)(x) = E(\varphi \cdot \psi)(x)$, as desired. It follows that E is indeed an embedding, concluding the proof. $\square$

(Anton Golov) Radboud University Nijmegen, Department of Mathematics, P.O. Box 9010, 6500 GL Nijmegen, the Netherlands
Email address: agolov@science.ru.nl

(Sebastiaan A. Terwijn) Radboud University Nijmegen, Department of Mathematics, P.O. Box 9010, 6500 GL Nijmegen, the Netherlands
Email address: terwijn@math.ru.nl