COMPUTABILITY IN PARTIAL COMBINATORY ALGEBRAS

SEBASTIAAN A. TERWIJN

Abstract. We prove a number of elementary facts about computability in partial combinatory algebras (pca's). We disprove a suggestion made by Kreisel about using Friedberg numberings to construct extensional pca's. We then discuss separability and elements without total extensions. We relate this to Ershov's notion of precompleteness, and we show that precomplete numberings are not 1–1 in general.

§1. Introduction. Combinatory algebra was founded by Schönfinkel [24] and Curry [10], and is closely related to the lambda calculus (cf. Barendregt [2]). Curry attempted to use combinatory algebra as a foundation of mathematics, and Church tried the same for the lambda calculus. Both attempts fell short (Church's system was inconsistent and Curry's was too weak), but the formalisms became important for other reasons, for example as foundational theories for the theory of computation. Partial combinatory algebra (pca) was first studied in Feferman [13] as an axiomatic approach to the theory of computation and the study of various constructive theories. See Troelstra and van Dalen [28] for a discussion of pca and the relation with constructive mathematics, as well as a varied list of models of pca. In this paper we discuss computability in pca's and relate this to the theory of numberings.

The work in this paper is related to several other approaches to abstract models of computation, many of which are discussed in Longley and Normann [20]. In particular there is the notion of a *Basic Recursive Function Theory* (BRFT), introduced by Wagner and Strong, which is closely related to Moschovakis' notion of a precomputation theory (cf. Odifreddi [21, p. 222]). Every BRFT gives rise to a pca, as pointed out in [9, p. 199]. This will be relevant below when we discuss the work of Kreisel.

Kreisel [17] eloquently discusses some of the reasons one might want to generalize the setting of classical computability theory. Kreisel's ideas were highly influential, cf. Sacks [23], and also the review by Yates [32]. Hyland wrote [15] as a kind of tribute to [17]. In Section 6 we disprove a suggestion

made by Kreisel in [17] (quoted as a theorem in Odifreddi [21]) about
constructing extensional models, using Friedberg's result that the partial
computable functions are computably enumerable without repetitions. We
prove that such a construction is impossible.

Cockett and Hofstra [9] discuss category theoretic approaches to com-
putability theory and pca's. They introduce the notion of a Turing category,
relaxing restrictions in earlier work on so-called recursion categories by
Di Paola and Heller, and then proceed to show that the study of Turing
categories is essentially (in a precise sense) equivalent to the study of
pca's.

The paper is organized as follows. In Section 2 we list some preliminaries
about pca's, and in Section 3 we discuss basics of computable and
computably enumerable (c.e.) sets in pca's. In Section 4 we prove that
Post's theorem fails in Kleene's second model $\mathcal{K}_2$. In Section 5 we discuss
the halting problem and relativization. In Section 6 we discuss Kreisel's
suggestion about Friedberg numberings and extensional pca's, and show
that it is impossible. In Section 7 we discuss inseparable sets, and in
Section 8 we use this in the discussion of elements without total extensions. In
Section 9 we relate this to the theory of numberings, and show that
precomplete generalized numberings cannot be 1–1. In Section 10 we
use the notion of relativization to formulate an analog of Arslanov's
completeness criterion for pca's, motivated by results about the theory of
numberings.

Our notation from computability theory is mostly standard and follows
Odifreddi [21] and Soare [26]. The natural numbers are denoted by ω.
$\omega^{<\omega}$ is the set of finite sequences over ω. φ_e, $e \in \omega$, denotes a standard
enumeration of the (unary) partial computable (p.c.) functions. Notation
for pca's is introduced in Section 2. For a partial function φ, $\mathrm{dom}(\varphi)$ denotes
the set of elements where φ is defined, and $\varphi \restriction x$ denotes its restriction to
numbers $n < x$.

§2. Partial combinatory algebras.

DEFINITION 2.1. A *partial applicative structure* (pas) is a set $\mathcal{A}$ together
with a partial map $\cdot$ from $\mathcal{A} \times \mathcal{A}$ to $\mathcal{A}$. We also write ab instead of $a \cdot b$, and
think of this as 'a applied to b'. If this is defined we denote this by $ab\downarrow$. By
convention, application associates to the left. We write abc instead of $(ab)c$.
Terms over $\mathcal{A}$ are built from elements of $\mathcal{A}$, variables, and application. If
t_1 and t_2 are terms then so is $t_1 t_2$. If $t(x_1, ..., x_n)$ is a term with variables
x_i, and $a_1, ..., a_n \in \mathcal{A}$, then $t(a_1, ..., a_n)$ is the term obtained by substituting
the a_i for the x_i. For closed terms (i.e., terms without variables) t and s, we
write $t \simeq s$ if either both are undefined, or both are defined and equal. Here
application is *strict* in the sense that for $t_1 t_2$ to be defined, it is required that
both t_1, t_2 are defined. We say that an element $f \in \mathcal{A}$ is *total* if $f a\downarrow$ for every
$a \in \mathcal{A}$.

A pas $\mathcal{A}$ is *combinatory complete* if for any term $t(x_1, ..., x_n, x)$, $0 \leqslant n$,
with free variables among $x_1, ..., x_n, x$, there exists a $b \in \mathcal{A}$ such that for all
$a_1, ..., a_n, a \in \mathcal{A}$,

(i) $ba_1 \cdots a_n \downarrow$,
(ii) $ba_1 \cdots a_n a \simeq t(a_1, ..., a_n, a)$.

A pas $\mathcal{A}$ is a *partial combinatory algebra* (pca) if it is combinatory complete.

Note that combinatory completeness is the analog of the S-m-n-theorem (also called the parametrization theorem) from computability theory, cf. Odifreddi [21].

THEOREM 2.2 (Feferman [13]). *A pas $\mathcal{A}$ is a pca if and only if it has elements k and s with the following properties for all $a, b, c \in \mathcal{A}$:*

- *k is total and $kab = a$,*
- *$sab \downarrow$ and $sabc \simeq ac(bc)$.*

Note that k and s are nothing but partial versions of the familiar combinators from combinatory algebra. As noted in [13, p. 95], Theorem 2.2 has the consequence that in any pca we can define lambda-terms in the usual way (cf. Barendregt [2, p. 152]):[1] For every term $t(x_1, ..., x_n, x)$, $0 \leqslant n$, with free variables among $x_1, ..., x_n, x$, there exists a term $\lambda^* x.t$ with variables among $x_1, ..., x_n$, with the property that for all $a_1, ..., a_n, a \in \mathcal{A}$,

- $(\lambda^* x.t)(a_1, ..., a_n) \downarrow$,
- $(\lambda^* x.t)(a_1, ..., a_n)a \simeq t(a_1, ..., a_n, a)$.

The most famous examples of a pca are Kleene's first and second models $\mathcal{K}_1$ and $\mathcal{K}_2$. $\mathcal{K}_1$ consists of the natural numbers ω, with application defined as $n \cdot m = \varphi_n(m)$. So this is essentially the setting of classical computability theory. $\mathcal{K}_2$ was introduced in [16], and is defined on ω^ω, with application $\alpha \cdot \beta$ defined by applying the continuous functional with code α to the real β. See Longley and Normann [20] for more details. Many other examples of pca's can be found in the books by Beeson [6], Odifreddi [21], and van Oosten [30].

The presence of the λ^*-terms and the combinators allows for the following definitions in any pca (cf. Barendregt [2, p. 44] and van Oosten [30]): The Booleans true and false can be defined as $\mathsf{T} = \lambda^* xy.x = k$ and $\mathsf{F} = \lambda^* xy.y = ki$, where $i = skk$. We can implement definition by cases using an element if-then-else with the property if-then-else$\mathsf{T}ab = a$ and if-then-else$\mathsf{F}ab = b$. Namely, we can simply take if-then-else $= i$. This also gives the Boolean operations, for example

$$\mathsf{not}a = \text{if-then-else}a\mathsf{FT}, \text{ and}$$
$$\mathsf{and}ab = \text{if-then-else}a(\text{if-then-else}b\mathsf{TF})\mathsf{F}.$$

Coding of sequences is a standard device in the lambda calculus. Using the λ^*-terms available in any pca, we can code n-tuples $(a_1, ..., a_n)$ by $\langle a_1, ..., a_n \rangle = \lambda^* z.za_1 ... a_n$. The inverse projection functions can be defined as $\mathsf{U}_i^n = \lambda^* u_1 ... u_n.u_i$, so that

---

[1]Because the lambda-terms in combinatory algebra do not have the same substitution properties as in the lambda calculus, we use the notation λ^* rather than λ, cf. Barendregt [2, p. 152]. E.g. [20, p. 84] the terms $\lambda x.(\lambda y.y)x$ and $\lambda x.x$ are β-equivalent, but their λ^*-versions are $s(ki)i$ and i, and these are in general different elements in a pca.

$$\langle a_1, ..., a_n \rangle \mathsf{U}_i^n = a_i.$$

There are various ways to define the natural numbers $\bar{0}, \bar{1}, \bar{2}, ...$ in a pca. A convenient way is to define $\bar{0} = i$, and $\overline{n+1} = \langle \mathsf{F}, \bar{n} \rangle$, cf. Barendregt [2, p. 44].

All the above can be defined in any pca, but they may trivialize if $|\mathcal{A}| = 1$. van Oosten [30, p. 11] calls $\mathcal{A}$ *nontrivial* if $|\mathcal{A}| > 1$. We note that $n = 1$ is the only possible cardinality for a finite pca:

PROPOSITION 2.3. *Suppose that a pca $\mathcal{A}$ is finite. Then $|\mathcal{A}| = 1$.*

PROOF. Note that every pca is nonempty, since by Feferman's Theorem 2.2 it has to contain the combinators k and s. Furthermore, there exists a (total) pca with precisely one element a, with application $aa \downarrow = a$. In this pca we have $s = k = a$. Since all λ^*-terms are equal to a, also $\mathsf{T} = \mathsf{F} = a$.

Now suppose that $\mathcal{A} = \{a_1, ..., a_n\}$, and $n > 1$. $\mathcal{A}$ contains the elements $ka_1, ..., ka_n$, which are n distinct constant functions since $ka_i b = a_i$. $\mathcal{A}$ also contains the identity function $i = skk$, which is not a constant function since $n > 1$. So $\mathcal{A}$ has at least $n+1$ elements, a contradiction. ⊣

Following [29], we say that a partial function $\varphi : \mathcal{A} \to \mathcal{A}$ is *representable* in $\mathcal{A}$ if there is an element $r \in \mathcal{A}$ such that for every $a \in \text{dom}(\varphi)$, $ra \downarrow = \varphi(a)$. We have a similar definition for multivariate functions.

§3. Computable sets and c.e. sets in pca's. The following definition is taken from van Oosten and Voorneveld [31], which in turn is based on Longley [19].

DEFINITION 3.1. Let $\mathcal{A}$ be a pca, and $A \subseteq \mathcal{A}$. A is called *decidable* in $\mathcal{A}$ if there exists a total $c \in \mathcal{A}$ such that for every $a \in \mathcal{A}$,

$$ca = \mathsf{T} \Longleftrightarrow a \in A,$$
$$ca = \mathsf{F} \Longleftrightarrow a \notin A.$$

Note that this is equivalent to saying that the characteristic function $\chi_A : \mathcal{A} \to \{\mathsf{T}, \mathsf{F}\}$ of A is representable in $\mathcal{A}$. Instead of T and F, we may equivalently use 0 and 1, cf. Proposition 3.4.

We can also easily define the analog of c.e. sets in the following way.

DEFINITION 3.2. We say that A is *computably enumerable* (*c.e.*) in $\mathcal{A}$ if there exists $e \in \mathcal{A}$ such that

$$A = \text{dom}(e) = \{a \in \mathcal{A} \mid ea \downarrow\}.$$

Note that this notion is not very useful in total pca's, since there $\mathcal{A}$ itself is the only c.e. set.[2] For nontotal pca's we have the following result.

PROPOSITION 3.3. *In nontotal pca's, decidable sets are c.e. The converse implication does not hold in general.*

---

[2]In total pca's, such as the lambda calculus, one can represent 'undefined' in other ways, for example using terms without normal form, cf. Barendregt [3].

PROOF. Suppose that $\mathcal{A}$ is a nontotal pca, and that $A \subseteq \mathcal{A}$ and $c \in \mathcal{A}$ are as in Definition 3.1 above. First note that $\mathcal{A}$ contains a totally undefined function. Namely, since $\mathcal{A}$ is nontotal, there are $f, g \in \mathcal{A}$ such that $fg\uparrow$. Now define $h = \lambda^* x. fg = s(kf)(kg)$. Then $ha\uparrow$ for every $a \in \mathcal{A}$.

Now define $ea = \text{if-then-else}(ca)0(ha)$. This yields 0 if $ca = \mathsf{T}$, so if $a \in A$, and ha, which is undefined, otherwise. Hence $ea\downarrow$ if and only if $a \in A$.

For the converse implication, c.e. sets are not always decidable by Proposition 5.1.   ⊣

Definition 3.1 uses T, F as truth values. In mathematics it is customary to use $0, 1$ as values of characteristic functions. We show that in the context of pca's, we may equivalently use the numerals $\bar{0}, \bar{1}$ as truth values.

PROPOSITION 3.4. *Let $\mathcal{A}$ be a pca. There exists $c, d \in \mathcal{A}$ such that*

$$
\begin{aligned}
c\mathsf{T} &= \bar{1} & d\bar{1} &= \mathsf{T} \\
c\mathsf{F} &= \bar{0} & d\bar{0} &= \mathsf{F}.
\end{aligned}
$$

PROOF. It is easy to check that $c = \lambda^* z.\text{if-then-else} z\bar{1}\bar{0}$ satisfies the first part of the proposition.

For the second part, note that the term $\text{zero} = \lambda^* x.x\mathsf{T}$ has the property $\text{zero}\bar{0} = \mathsf{T}$ and $\text{zero}\bar{1} = \mathsf{F}$ (cf. [2, p. 134]) so we can take d to be the term $\lambda^* x.\text{not}(\text{zero} x)$.   ⊣

Since in every pca there are elements mapping T and F to $\bar{1}$ and $\bar{0}$, and vice versa, we may equivalently use $\bar{1}$ and $\bar{0}$ in Definition 3.1. From now on we will mostly use the latter, and simply write 0 and 1 for the values of characteristic functions.

§4. A counterexample to Post's theorem. Post's theorem is the statement that for $A \subseteq \omega$, if both A and its complement $\overline{A}$ are c.e., then A is decidable. To decide whether $x \in A$, simply enumerate both A and $\overline{A}$ until x appears in one of them. This works because in ω, c.e. sets have finite approximations, and if $x \in A$ then x appears in A after finitely many steps. In general, we do not have a good notion of approximation in pca's, and being "enumerated" into a c.e. set does not have to happen in finitely many stages in every pca. Hence there does not seem to be a reason why Post's theorem should hold in general. Indeed we now show that it fails in Kleene's second model $\mathcal{K}_2$.

PROPOSITION 4.1. *Post's theorem fails in $\mathcal{K}_2$.*

PROOF. Let $\bar{0}$ denote the all zero sequence in 2^ω, and let $\bar{1}$ denote the all one sequence.

Let $A = \{\bar{0}\}$. Then A is c.e. in $\mathcal{K}_2$: Define $\hat{\alpha} : 2^{<\omega} \to 2^\omega$ by

$$
\hat{\alpha}(x) = \begin{cases} 0^n & \text{if } x = 0^n, \\ \uparrow & \text{otherwise.} \end{cases}
$$

$\hat{\alpha}$ defines a partial computable functional $\alpha : 2^\omega \to 2^\omega$ that simply copies the input, as long as the input consists of only zeros, and becomes undefined otherwise. Hence $\text{dom}(\alpha) = \bar{0}$, which shows that A is c.e.

The complement $\overline{A} = \{r \in 2^\omega : r \neq \bar{0}\}$ is also c.e.: Define

$$\hat{\beta}(x) = \begin{cases} 1^n & \text{if } n = |x| \text{ and } x(i) = 1 \text{ for some } i < n, \\ \uparrow & \text{otherwise.} \end{cases}$$

Then $\hat{\beta}$ defines a partial computable functional β with $\text{dom}(\beta) = \overline{A}$.

So both A and $\overline{A}$ are c.e., but A is not decidable in $\mathcal{K}_2$. Suppose that it were, and suppose that γ is a computable functional such that $\gamma(x) = \mathsf{T}$ if $x = \bar{0}$, and $\gamma(x) = \mathsf{F}$ if $x \neq \bar{0}$. Now since γ is continuous, this distinction has to be made on the basis of a finite initial segment of x, which is impossible. Note that for this argument it does not really matter what T and F are, as long as they are distinct reals in 2^ω. ⊣

§5. Reductions and relativization. In Beeson [6, p. 107] it is already remarked that besides the existence of a universal function and the undecidability of the halting problem, not many analogues of classical results in computability theory can be proved. We claim no originality for the results in this section, but for the record discuss the m-completeness of the halting problem. This was surely known to people working in axiomatic recursion theory, but since we have not been able to locate it in the literature, we include it here. In any case, it is an easy fact that is completely analogous to Turing's classical result.

We can define the analog of the halting problem in any pca $\mathcal{A}$ using the coding of sequences. Define

$$H = \{\langle a,b \rangle \mid ab\downarrow\}.$$

PROPOSITION 5.1. *For every nontotal pca $\mathcal{A}$, H is undecidable and c.e. in $\mathcal{A}$.*

PROOF. To see that H is c.e. in $\mathcal{A}$, define $e \in \mathcal{A}$ by

$$ex = (x\mathsf{U}_1^2)(x\mathsf{U}_2^2),$$

where U_i^2 refers to the projection functions defined in Section 2. We then have in particular that

$$e\langle a,b \rangle \downarrow \iff ab\downarrow \iff \langle a,b \rangle \in H,$$

which shows that H is indeed c.e. in $\mathcal{A}$.

The proof of the undecidability is the same as for the classical case. Namely suppose that H were decidable. This would mean the existence of $f \in \mathcal{A}$ such that

$$f\langle a,b \rangle = \begin{cases} \mathsf{T} & \text{if } ab\downarrow, \\ \mathsf{F} & \text{if } ab\uparrow. \end{cases}$$

Define $g \in \mathcal{A}$ such that $ga\downarrow$ if and only if $f\langle a,a \rangle = \mathsf{F}$. Such g can be defined using the if-then-else operator (cf. Section 2) as follows. Let $ha\uparrow$ for every a. Such h exists in any nontotal pca, cf. the proof of Proposition 3.3. Now define

$$ga = \mathsf{if\text{-}then\text{-}else}(\mathsf{not}(f\langle a,a \rangle))0(ha).$$

This yields 0 if $f\langle a,a\rangle = \mathsf{F}$, hence if $aa\uparrow$, and ha, which is undefined, otherwise. Hence $ga\downarrow$ if and only if $aa\uparrow$. Taking $a = g$ we obtain a contradiction. $\dashv$

Note that by Proposition 5.1, as soon as a pca has *one* undefined application $ab\uparrow$, its halting problem is undecidable.

Defining the analog of m-reductions is also straightforward:

DEFINITION 5.2. For sets $A, B \subseteq \mathcal{A}$, we say that A *m-reduces* to B, denoted $A \leqslant_m B$, if there exists a total element $f \in \mathcal{A}$ such that

$$a \in A \Longleftrightarrow fa \in B,$$

for every $a \in \mathcal{A}$. We write $A \equiv_m B$ if both $A \leqslant_m B$ and $B \leqslant_m A$, in which case we say that A and B have the same *m-degree*.

Many basic properties of m-reductions in ω carry over to the general case. For example, we can define the *diagonal halting problem*

$$K = \{a \mid aa\downarrow\},$$

and show that $K \equiv_m H$.

PROPOSITION 5.3. *H is m-complete for the c.e. sets in $\mathcal{A}$, i.e., $A \leqslant_m H$ for every such set A.*

PROOF. Suppose that A is c.e. in $\mathcal{A}$, say $A = \mathrm{dom}(e)$ for $e \in \mathcal{A}$. Then

$$e \in A \Leftrightarrow ea\downarrow \Leftrightarrow \langle e,a\rangle \in H,$$

and hence $fa = \langle e,a\rangle$ is an m-reduction from A to H. Note that f is total, since $\langle\cdot,\cdot\rangle$ is implemented by λ^*-terms in any pca. $\dashv$

We also have an analog of Turing reductions in any pca $\mathcal{A}$. This is somewhat harder to define, and was carried out in van Oosten [29], see also [31]. This gives for any pca $\mathcal{A}$ and any partial function $f : \mathcal{A} \to \mathcal{A}$ a new pca $\mathcal{A}[f]$ in which f is represented, in such a way that $\mathcal{A}[f]$ contains $\mathcal{A}$ in a natural way. Application in $\mathcal{A}[f]$ models computation in $\mathcal{A}$ with f as an oracle, and thus provides an analog of relativization for pca's. For $\mathcal{A} = \mathcal{K}_1$ we have that $A \in \mathcal{K}_1[B]$ is equivalent to Turing reducibility $A \leqslant_T B$.

More specifically, $\mathcal{A}[f]$ has the same underlying set as $\mathcal{A}$, with only a different application operator $\cdot_f$, defined as follows. $a \cdot_f b\downarrow = c$ if there exist $e_0, ..., e_{n-1} \in \mathcal{A}$ (the queries to the oracle f) such that for every $i < n$:

- $a \cdot \langle b, f(e_0), ..., f(e_{i-1})\rangle = \langle \mathsf{F}, e_i\rangle$,
- $a \cdot \langle b, f(e_0), ..., f(e_{n-1})\rangle = \langle \mathsf{T}, c\rangle$.

Here $\cdot$ denotes application in $\mathcal{A}$. We will use this construction in Sections 8 and 10.

§6. Extensionality and enumerations without repetitions.

DEFINITION 6.1. A pca $\mathcal{A}$ is called *extensional* if

$$\forall a \in \mathcal{A}(fa \simeq ga) \Longrightarrow f = g,$$

for all $f, g \in \mathcal{A}$.

In [14] Friedberg proved the classic result that the class of partial computable functions is c.e. without repetitions. In the second edition of Odifreddi [21, p. 224] it is stated that the existence of an extensional partial combinatory algebra follows from Friedberg's result, with a reference to Kreisel [17]. (This is from the second edition of [21]; in the first edition this statement is missing.) Indeed, Kreisel (p186 ibid.) suggested that an enumeration without repetitions could be used to obtain an extensional model. (Note however that Kreisel's concern were models of BRFT mentioned above, not pca's.) However, Kreisel explicitly says that he did not verify this result.[3] In any case, it seems that the mere statement of Friedberg's theorem is not sufficient to obtain an extensional pca, so that at least an adaptation of the proof of Friedberg's result is required.

Suppose that ψ_e, $e \in \omega$, is an enumeration of all unary p.c. functions. On the face of it, it seems plausible that one could make this into an extensional pca, since after all every function in the enumeration has a unique code. Of course the intended application operator here is

$$n \cdot m = \psi_n(m). \tag{1}$$

To prove that ω with this application operator is a pca, one has to show that there exist combinators k and s as in Theorem 2.2. Now the statement of Friedberg's theorem itself is not sufficient to prove this. Namely, for every a there is a code ka of the constant a function in the enumeration, but we cannot find such codes uniformly in a. Hence we cannot prove that we have a combinator k satisfying $kab = a$ for every a and b, which is the first requirement of Theorem 2.2. This obstacle, however, can be overcome by an adaptation of the proof of Friedberg's result, see Theorem 6.3(i). However, for the combinator s no such adaptation is possible.

LEMMA 6.2. *There exists a computable enumeration without repetitions ψ_x of the unary p.c. functions such that for all $x, y \in \omega$,*

$$\psi_{2x+1}(y) = x. \tag{2}$$

PROOF. For the application of the lemma below, we need to be able to effectively retrieve the combinator k (as in Theorem 2.2) from the enumeration. Note that for every a, ka is the function that is constant a. We code the functions ka on the odd numbers by defining ψ_{2a+1} as in (2). We use the even numbers for the construction of the enumeration of all other p.c. functions, in the manner of Friedberg [14] (see also Odifreddi [21, II.5.22]).[4]

---

[3]Kreisel discusses the relation between BRFT and set theory, using generalized recursion theory. In this context he discusses extensionality. Kreisel writes: "...there are two ways of treating *extensionality*. [The first is proof-theoretic] Another is to appeal to an *enumeration without repetition*; but I have not stopped to verify the obvious essential point whether the axioms of BRFT are in fact satisfied for such an enumeration without repetition..."

[4]There is nothing very special about the class of constant functions being fixed in this lemma. Similar modifications of Friedberg's result have been made by Pour-El and Howard and others, cf. [21, p. 232] for references and further discussion.

We start the construction by fixing ψ_{2x+1} as in (2) for every x. We construct ψ_{2x} in such a way that every unary p.c. function occurs exactly once. (We assume that φ_e is an enumeration of all unary p.c. functions.) We do this by letting ψ_{2x} follow some φ_e for every x. If subsequently it looks like e is not a minimal code of φ_e, or that φ_e is one of the constant functions, we release the follower by making ψ_{2x} a finite function different from all functions occurring so far, and stop its enumeration.

We say that x is a *follower* of φ_e at stage s if we are trying to make $\psi_x = \varphi_e$, i.e., $\psi_{x,s} = \varphi_{e,s}$. A follower x of φ_e is *permanent* if it is a follower of φ_e at almost every stage.

For a follower x of φ_e, to *release* x at stage s means that x is no longer a follower of φ_e, and that we define ψ_x to be different from all other finite functions $\psi_{y,t}$ that have been defined so far, i.e., with $y \neq x$, $t \leqslant s$, and with either $0 < y < s$ or y odd, by making it a finite function incompatible to these. We will have that ψ_0 is the empty function, and all other ψ_x will have nonempty domain. Since we work in $\omega^{<\omega}$ it will always be possible to find incompatible strings for finite functions with nonempty domain. To make ψ_x incompatible with all ψ_y with y odd, it suffices to make it nonconstant.

The formal construction is as follows. At stage $s = 0$, define ψ_{2x+1} as in (2) for every x. Also define $\psi_0 = \emptyset$ to be the empty function. In the rest of the construction, we only consider φ_e with nonempty domain.

At stage $s > 0$ of the construction we do the following.

For every $x < s$, if x is a follower of φ_e, we release x if x is even and one of the following holds:

- there is $i < e$ with $\varphi_{i,s} \upharpoonright x = \varphi_{e,s} \upharpoonright x$. (In this case e does not look like a minimal code.)
- for some follower y already released, $\psi_{y,s} = \psi_{x,s}$. (In this case φ_e might equal the finite function ψ_y, and we have to avoid the duplication.)
- $\varphi_{e,s} \upharpoonright x$ is a constant function, i.e., $\exists a \forall n < x (\varphi_{e,s}(n) = a)$. (Since the constant functions are already covered by the ψ_{2x+1}.)

If $s = \langle e, t \rangle$, and $\varphi_{e,s} \neq \emptyset$, and φ_e currently does not have a follower, pick the smallest even x that has not yet been used as a follower, and appoint x as a follower of φ_e. Note that this ensures that every nonempty φ_e has infinitely many opportunities of being appointed a follower.

Finally, for every x and e such that x is a follower of φ_e at stage s, define $\psi_{x,s} = \varphi_{e,s}$. This ends the construction. We verify that the enumeration ψ_x is as desired.

Claim: $\forall e \exists x (\varphi_e = \psi_x)$, i.e., every unary p.c. function occurs in the enumeration ψ_x. To prove the claim, suppose that e is a minimal index of φ_e, and that s_0 is so large that

$$\forall s > s_0 \, \forall x > s_0 \, \forall i < e \, (\varphi_{i,s} \upharpoonright x \neq \varphi_{e,s} \upharpoonright x).$$

If φ_e is constant then it is equal to ψ_{2x+1} for some x by stage 0 of the construction. Suppose that φ_e is not constant. If φ_e has a permanent follower x then $\varphi_e = \psi_x$. Otherwise, φ_e keeps getting appointed new followers (at stages of the form $s = \langle e, t \rangle$). Since the first option for releasing a follower x

is ruled out after stage s_0 by assumption, and the third is ruled out because φ_e is not constant, the only option for releasing x after this stage is the second one, namely that $\psi_{y,s} = \psi_{x,s}$ for some already released y. But this can happen only once, since all $\psi_{y,s}$ for y released are incompatible.

Claim: $x \neq y \implies \psi_x \neq \psi_y$, i.e., ψ_x is an enumeration without repetitions. Namely, the ψ_x for x odd are all different by (2). Note further that every even $x > 0$ is eventually used as a follower, since we always pick the smallest one not used yet.

If x is a follower of φ_e, and φ_e is constant, then x will eventually be released by the third reason for release in the construction. Hence ψ_x is never a constant function for any even x. So it suffices to prove the claim for x and y even. We check the following cases.

Suppose that x and y are permanent followers, say $\psi_x = \varphi_e$ and $\psi_y = \varphi_i$. Then $e \neq i$ since φ_e can have at most one permanent follower. W.l.o.g. suppose $i < e$. If $\psi_x = \psi_y$ then $\varphi_e = \varphi_i$, hence there is a stage s such that $\varphi_{i,s} \upharpoonright x = \varphi_{e,s} \upharpoonright x$, causing x to be released, contrary to assumption. Hence we must have $\psi_x \neq \psi_y$.

Suppose that both x and y are released at some stage. Since all functions ψ_x and ψ_y for different released x and y are incompatible, this implies that they are different.

Finally suppose that one of x and y is permanent and the other is released, say x is permanent and y is not. If $\psi_x = \psi_y$, then ψ_x is a finite function. So at some stage s we will have $\psi_{y,s} = \psi_{x,s}$, causing x to be released, contradicting the assumption.

This proves the second claim, and the proof of the lemma. $\dashv$

THEOREM 6.3.   (i) *There exists an extensional pas on the set of all partial computable unary functions, containing a combinator k as in Theorem 2.2.*

(ii) *There exists no such a pas with the combinator s.*

PROOF.   (i) Using the enumeration ψ_x from Lemma 6.2, define application as in (1). This pas is clearly extensional, as the enumeration ψ_x is 1–1. We show that we have the combinator k. Let k be a code such that $\psi_k(a) = 2a + 1$ for every a. The code k exists because this is a computable function, so it occurs in the enumeration. Then by (2) we have

$$\psi_{\psi_k(a)}(b) = a,$$

hence k is a code of a total function with the property $kab = a$ for every a.

For the proof of part (ii), suppose that ψ_e, $e \in \omega$, is a computable enumeration without repetitions containing all unary p.c. functions, and suppose that ω with the application operator (1) is a pca. We make the following observations.

I. $\{a \in \omega : \psi_a(a)\downarrow\}$ is undecidable. This is the same as the usual argument for the undecidability of the halting problem: Suppose that c is a code such that $\varphi_c(a)\downarrow \Leftrightarrow \psi_a(a)\uparrow$ for every a. Since the enumeration ψ_e contains a code of every p.c. function, there exists e such that $\psi_e = \varphi_c$. Taking $a = e$ we obtain a contradiction: $\psi_e(e)\downarrow \Leftrightarrow \varphi_c(e)\downarrow \Leftrightarrow \psi_e(e)\uparrow$.

II. $\{b \in \omega : \psi_b \text{ is constant zero}\}$ is decidable.[5] Suppose c is a code such that ψ_c is the constant zero function. Since codes in the enumeration ψ_e are unique, ψ_b is constant zero if and only if $b = c$.

Since we have assumed that ω with application (1) is a pca, we have combinatory completeness (see Definition 2.1), which is an analogue of the S-m-n-theorem. Using this we can reduce I to II, and thus we obtain a contradiction. Namely, consider the term

$$t(x,y) = 0 \cdot (x \cdot x).$$

Here $0 \cdot x$ should be read as the constant zero function applied to x (which happens to be the same notation as multiplying with 0). By combinatory completeness, there exists $f \in \omega$ such that for every a and c in $\mathcal{A}$, $fa \downarrow$ and $fac \simeq 0(aa) = 0 \cdot \psi_a(a)$. So we have that $\psi_a(a) \downarrow$ if and only if ψ_{fa} is the constant zero function. Because $fa = \psi_f(a)$ is a total computable function, this constitutes an m-reduction from I to II. Since the set from II is decidable, it follows that the one from I is also decidable, contradicting what we proved above.   ⊣

COROLLARY 6.4. *There does not exist an extensional pca on the set of all p.c. functions* (*with application the intended one*).

PROOF. By Theorem 2.2, such a pca would have to contain combinators s and k, which is impossible by Theorem 6.3(ii).   ⊣

As we mentioned above, Kreisel's suggestion was about models of BRFT, not pca's. However, since every BRFT gives rise to a pca (cf. the introduction), Corollary 6.4 also precludes the use of Friedberg's result to construct extensional models of BRFT.

§7. Inseparability. In this section we show that every pca has computably inseparable subsets A and B. This is completely analogous to the situation in classical computability theory, even though the sets A and B may not always be representable in the pca. We use this in the following sections when we discuss elements without total extensions.

DEFINITION 7.1. Let $\mathcal{A}$ be a pca. We call a pair of disjoint subsets $A, B \subseteq \mathcal{A}$ *computably separable* if there exists a decidable subset $C \subseteq \mathcal{A}$ such that $A \subseteq C \subseteq \overline{B}$, and *computably inseparable* otherwise.

Define

$$A = \{a \in \mathcal{A} \mid aa \downarrow = 0\},$$
$$B = \{a \in \mathcal{A} \mid aa \downarrow = 1\}.$$

PROPOSITION 7.2. *The sets A and B are computably inseparable in $\mathcal{A}$.*

---

[5] Note that for the standard numbering φ_e of the p.c. functions, the set from II is Π_2^0-complete.

PROOF. Suppose that $A \subseteq C \subseteq \overline{B}$ and that C is decidable by $c \in \mathcal{A}$. Then

$$c \in C \implies cc \downarrow = 1 \implies c \in B \implies c \notin C,$$
$$c \notin C \implies cc \downarrow = 0 \implies c \in A \implies c \in C,$$

and we have a contradiction. ⊣

Note that for $\mathcal{K}_1$, the sets A and B are the standard example of a pair of computably inseparable c.e. sets. We note that the sets A and B need not always be c.e. in $\mathcal{A}$. A sufficient condition for A and B to be c.e. is that every singleton $\{a\}$ is c.e. in $\mathcal{A}$. To see that this implies that A is c.e., suppose that $e \in \mathcal{A}$ is such that $ea \downarrow \Leftrightarrow aa \downarrow$. Since $\{0\}$ is c.e., there exists $d \in \mathcal{A}$ such that $da \downarrow \Leftrightarrow a = 0$. Then we have

$$d(ea) \downarrow \Leftrightarrow ea \downarrow = 0 \Leftrightarrow aa \downarrow = 0,$$

hence A is c.e. in $\mathcal{A}$. The condition that every singleton is c.e. holds in $\mathcal{K}_1$ and $\mathcal{K}_2$.

Scott (cf. [2, Theorem 6.6.2]) proved that for the set of terms Λ in the lambda calculus, any pair of disjoint subsets that are closed under equality is computably inseparable. Note that this refers to ordinary computable inseparability in ω, using a suitable coding of lambda-terms [2, Definition 6.5.6]. Note that Definition 7.1 is more general, as it also applies to uncountable domains.

§8. Elements without total extensions.

DEFINITION 8.1. For elements b and f of a pca $\mathcal{A}$, we say that f is a *total extension* of b if f is total and for every $a \in \mathcal{A}$,

$$ba \downarrow \implies fa = ba.$$

It is well-known that there exist p.c. functions without total computable extensions. This follows e.g. from the existence of computably inseparable c.e. sets. The existence of inseparable sets from Proposition 7.2 does not immediately yield the same result for pca's, as these sets do not have to be c.e. in $\mathcal{A}$. To obtain elements without total extensions, an extra property is needed.

DEFINITION 8.2. We say that 0, 1 are *separable* in $\mathcal{A}$ if there exists a total 0–1-valued $c \in \mathcal{A}$ such that for every $a \in \mathcal{A}$,

$$ca = 0 \implies a \neq 1,$$
$$ca = 1 \implies a \neq 0.$$

Note that separability of 0, 1 in $\mathcal{A}$ implies that $0 \neq 1$, and that it is equivalent to the statement that the subsets $\{0\}$, $\{1\}$ are computably separable. This provides a *constructive* way to verify for every element $a \in \mathcal{A}$ the formula $a \neq 0 \lor a \neq 1$.

In Definition 8.2 we have used 0 and 1, i.e., the numerals $\bar{0}$ and $\bar{1}$ (cf. the discussion in Section 3), but the notion of separability would apply to any other pair of elements from $\mathcal{A}$.

Separability of 0 and 1 is satisfied in $\mathcal{K}_1$ and $\mathcal{K}_2$, but not in every pca. For example, it does not hold in the λ calculus. By Corollary 8.4 below, 0 and 1 are inseparable in any total pca, and by Theorem 8.5 there also exist nontotal examples where this is the case.

THEOREM 8.3. *Suppose that $\mathcal{A}$ is a pca such that* 0, 1 *are separable in $\mathcal{A}$. Then there exists $b \in \mathcal{A}$ without a total extension $f \in \mathcal{A}$.*

PROOF. Define $ba = aa$. (Note that such a b exists by combinatory completeness applied to the term $t(x) = xx$.) Suppose that $f \in \mathcal{A}$ is a total extension of b, and let $c \in \mathcal{A}$ be a total 0–1-valued separation of 0, 1 as in Definition 8.2. Then $\hat{f}a = c(fa)$ is also 0–1-valued, and again $\hat{f} \in \mathcal{A}$ by combinatory completeness. Now

$$aa\!\downarrow\,= 0 \Longrightarrow ba\!\downarrow\,= 0 \Longrightarrow fa = 0 \Longrightarrow \hat{f}a = c(fa) = 0,$$
$$aa\!\downarrow\,= 1 \Longrightarrow ba\!\downarrow\,= 1 \Longrightarrow fa = 1 \Longrightarrow \hat{f}a = c(fa) = 1,$$

and hence $\hat{f}$ is a total 0–1-valued extension of b. But this contradicts the computable inseparability of the sets A and B from Proposition 7.2.   ⊣

Note that the proof of Theorem 8.3 still does not require the sets A and B to be c.e. in $\mathcal{A}$.

COROLLARY 8.4. *In any total pca $\mathcal{A}$ (i.e., in any combinatorial algebra),* 0 *and* 1 *are inseparable.*

PROOF. If 0, 1 are separable in $\mathcal{A}$ then by Theorem 8.3 there exists an element without a total extension, which is clearly impossible if $\mathcal{A}$ is total.   ⊣

By Corollary 8.4, if 0, 1 are separable in $\mathcal{A}$ then $\mathcal{A}$ is not total. The converse of this does not hold by the next theorem.

THEOREM 8.5. *There exists a nontotal pca $\mathcal{A}$ in which* 0, 1 *are inseparable.*

PROOF. Let $\mathcal{A}$ be any nontrivial total pca, and let f be representable in $\mathcal{A}$. (We can simply take $f \in \mathcal{A}$.) We use the construction of the relativized pca $\mathcal{A}[f]$ from Section 5. By van Oosten [29, Corollary 2.3], the pca $\mathcal{A}[f]$ is never total, even if $\mathcal{A}$ is total. (This is due to the different interpretation of application in $\mathcal{A}[f]$, which allows for the definition of undefined functions.) It is easy to verify that $\mathcal{A}[f]$ has the same total functions as $\mathcal{A}$. (Simply replace queries to the oracle f by computations in $\mathcal{A}$.) So if 0 and 1 are separable in $\mathcal{A}[f]$ by a total 0–1-valued function c, then the same must hold in $\mathcal{A}$. But $\mathcal{A}$ is total, hence by Corollary 8.4, 0, 1 are inseparable in $\mathcal{A}$.   ⊣

Consider the following statements about a pca $\mathcal{A}$:

(i) 0, 1 are separable in $\mathcal{A}$.
(ii) The function $ba = aa$ has no total extension in $\mathcal{A}$.
(iii) There exists an element in $\mathcal{A}$ without total extension in $\mathcal{A}$.
(iv) $\mathcal{A}$ is not total.

We have (i) $\Rightarrow$ (ii) $\Rightarrow$ (iii) $\Rightarrow$ (iv): The first implication follows from the proof of Theorem 8.3, and the others are obvious. In fact, (ii) $\Leftrightarrow$ (iii), as can be seen as follows. The application function $d\langle a,b\rangle = ab$ is universal, so it suffices to prove that if $ba = aa$ has a total extension, then so has d. Suppose that f is a total extension of b, and let g be such that $g\langle a,b\rangle x \simeq ab$ for all a,b,x. (We can take $g = \lambda^* z,x.d(kzx)$.) Then

$$ab{\downarrow} \Longleftrightarrow g\langle a,b\rangle(g\langle a,b\rangle){\downarrow} = f(g\langle a,b\rangle),$$

so $f(g\langle a,b\rangle)$ is a total extension of d.

By Theorem 8.5 we have that (iv) $\nRightarrow$ (i), but we can in fact say more. In Section 6 we discussed Kreisel's suggestion for constructing a nontotal extensional pca from a Friedberg numbering. Despite the failure of this (Corollary 6.4), such pca's $\mathcal{A}$ do exist, as was proven in Bethke and Klop [7]. Since $\mathcal{A}$ is extensional, every element in $\mathcal{A}$ has a total extension in $\mathcal{A}$, as was proven in [5].[6] Since $\mathcal{A}$ is nontotal, we have (iv) $\nRightarrow$ (iii). At the moment we do not know whether (ii) $\nRightarrow$ (i).

The negation of item (iii) does not imply that $\mathcal{A}$ has a total completion in the sense of Bethke et al. [8], as one might think. Indeed, $\neg$(iii) implies that in particular the application function $d\langle a,b\rangle = ab$ has a total extension $h \in \mathcal{A}$, but this total extension h does not have to respect the structure of the combinator s. In fact, if we let $\mathcal{A}$ be nontotal and extensional as above, by extensionality $\neg$(iii) holds in $\mathcal{A}$ (cf. [5, Proposition 5.2]), but $\mathcal{A}$ is not completable by [8, Theorem 7.2].

§9. Precompleteness and 1–1 numberings.

In this section we consider numberings without repetitions, often simply called 1–1 *numberings*.

A numbering of the p.c. functions that is equivalent to the standard numbering is called *acceptable* [21, p. 215]. Rogers [22] showed that acceptable numberings are precisely those for which the enumeration theorem and parametrization (= the S-m-n-theorem) hold. It also follows from this that for any acceptable numbering the padding lemma holds, ensuring that every p.c. function has infinitely many codes. In particular, we see that no 1–1 numbering of the p.c. functions (such as Friedberg's numbering) is acceptable. For more on 1–1 numberings see Kummer [18].

A general theory of countable numberings was initiated by Ershov [11]. A numbering of a set S is simply a surjective function $\gamma : \omega \to S$. In particular, Ershov introduced the notion of a *precomplete* numbering on ω, and he proved in [12] that Kleene's recursion theorem holds for every precomplete numbering. Barendregt and Terwijn [4] extended the setting to partial combinatory algebra by defining the notion of a *generalized numbering* as a surjective function $\gamma : \mathcal{A} \to S$, where $\mathcal{A}$ is a pca and S is a set. The notion of precompleteness for generalized numberings was also defined in [4]. It is equivalent to the following definition:

---

[6]It follows from Proposition 5.2 in [5] that if $\mathcal{A}$ is extensional then the identity on $\mathcal{A}$ is precomplete, which is equivalent to the statement that every element in $\mathcal{A}$ has a total extension in $\mathcal{A}$.

DEFINITION 9.1. A generalized numbering $\gamma : \mathcal{A} \to S$ is *precomplete* if for every $b \in \mathcal{A}$ there exists a total element $f \in \mathcal{A}$ such that for all $a \in \mathcal{A}$,

$$ba\!\downarrow \implies fa \sim_\gamma ba. \tag{3}$$

In this case, we say that *f totalizes b modulo $\sim_\gamma$*.

Ershov's notion of precomplete numbering is obtained from this by taking for $\mathcal{A}$ Kleene's first model $\mathcal{K}_1$. Section 5 of [5] studies the relations between combinatory completeness, extensionality, and precompleteness of generalized numberings.

The standard numbering of the p.c. functions is precomplete by the S-m-n-theorem, and since every acceptable numbering is equivalent to the standard numbering it follows that acceptable numberings are precomplete. On the other hand, Friedberg's 1–1 numbering is not precomplete. We generalize this fact in Theorem 9.2 below.

Precompleteness is connected to the question which elements have total extensions, studied in Section 8. For example, the identity $\gamma_{\mathcal{A}} : \mathcal{A} \to \mathcal{A}$ is precomplete if and only if every element $b \in \mathcal{A}$ has a total extension $f \in \mathcal{A}$.

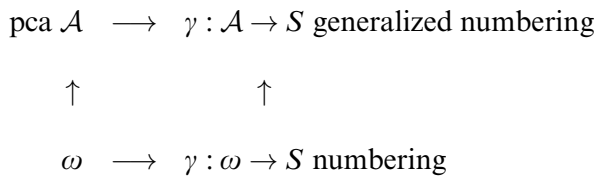THEOREM 9.2. *Suppose $\gamma : \mathcal{A} \to S$ is a precomplete generalized numbering, and that 0, 1 are separable in $\mathcal{A}$. Then γ is not 1–1.*

PROOF. Suppose that γ is precomplete and 1–1, and suppose that $b \in \mathcal{A}$. Since γ is precomplete, there exists $f \in \mathcal{A}$ that totalizes b modulo $\sim_\gamma$. As γ is 1–1 we have

$$ba\!\downarrow \implies fa \sim_\gamma ba \implies fa = ba$$

for every a. Hence every $b \in \mathcal{A}$ has a total extension $f \in \mathcal{A}$. But this contradicts Theorem 8.3. $\dashv$

§10. Arslanov's completeness criterion. Ershov [12] showed that Kleene's recursion theorem holds for any precomplete numbering $\gamma : \omega \to S$. Working in another direction, Feferman [13] proved that the recursion theorem holds in any pca $\mathcal{A}$. In Barendregt and Terwijn [4], the fixed point theorems of Ershov and Feferman were combined by proving a fixed point theorem for precomplete *generalized* numberings $\gamma : \mathcal{A} \to S$, that instead of ω have an arbitrary pca $\mathcal{A}$ as a basis. The following diagram summarizes the various possible settings of the recursion theorem.

$$\text{pca } \mathcal{A} \quad \longrightarrow \quad \gamma : \mathcal{A} \to S \text{ generalized numbering}$$

$$\uparrow \qquad\qquad\qquad \uparrow$$

$$\omega \quad \longrightarrow \quad \gamma : \omega \to S \text{ numbering}$$

Now another famous extension of the recursion theorem is Arslanov's completeness criterion [1], which extends the recursion theorem from computable functions to the class of all functions that are computable from a Turing-incomplete c.e. set. Explicitly, suppose that $A \subseteq \omega$ is a c.e. set such

that $K \not\leq_T A$, and suppose that f is an A-computable function. Then there exists $e \in \omega$ such that for all $x \in \omega$,

$$\varphi_{f(e)}(x) \simeq \varphi_e(x).$$

In Selivanov [25] it was shown that Arslanov's completeness criterion also holds for any precomplete numbering. (In contrast to this, it is open whether the joint generalization from Terwijn [27] also holds for every precomplete numbering.) This prompts the question whether Arslanov's completeness criterion also holds for generalized numberings. A first step would be to prove an analog of Arslanov's result for pca's. Using the concepts of Section 5, we can formulate such an analog as follows.

Let $\mathcal{A}$ be a pca, and suppose that A is c.e. in $\mathcal{A}$ such that $K \notin \mathcal{A}[A]$, where K is the halting set in $\mathcal{A}$ defined in Section 5. Note that this is the analog of stating that A is a c.e. set that is not Turing complete. Now Arslanov's result says that any A-computable function f has a fixed point, which translates to the following. Suppose that $f \in \mathcal{A}[A]$ is total. Then there exists $e \in \mathcal{A}$ such that for all $x \in \mathcal{A}$,

$$f \cdot_A e \cdot x \simeq e \cdot x.$$

Here $\cdot_A$ denotes application in $\mathcal{A}[A]$ and $\cdot$ denotes application in $\mathcal{A}$.

QUESTION 10.1. *Does this analog of Arslanov's completeness criterion hold for every pca?*

REFERENCES

[1] M. M. ARSLANOV, *On some generalizations of the fixed point theorem*. **Soviet Mathematics (Izvestiya VUZ. Matematika)**, vol. 25 (1981), no. 5, pp. 1–10 (English translation).

[2] H. P. BARENDREGT, **The Lambda Calculus**, second ed., Studies in Logic and the Foundations of Mathematics, vol. 103, North-Holland, Amsterdam, 1984.

[3] ———, *Representing 'undefined' in lambda calculus*. **Journal of Functional Programming**, vol. 2 (1992), no. 3, pp. 367–374.

[4] H. P. BARENDREGT and S. A. TERWIJN, *Fixed point theorems for precomplete numberings*. **Annals of Pure and Applied Logic**, vol. 170 (2019), pp. 1151–1161.

[5] ———, *Partial combinatory algebra and generalized numberings*, preprint, 2019, arXiv:1910.07750.

[6] M. J. BEESON, **Foundations of Constructive Mathematics**, Springer-Verlag, Berlin, 1985.

[7] I. BETHKE and J. W. KLOP, *Collapsing partial combinatory algebras*, **Higher-Order Algebra, Logic, and Term Rewriting** (G. Dowek et al., editors), Lecture Notes in Computer Science, vol. 1074, Springer, Berlin, 1996, pp. 57–73.

[8] I. BETHKE, J. W. KLOP, and R. DE VRIJER, *Extending partial combinatory algebras*. **Mathematical Structures in Computer Science**, vol. 9 (1999), pp. 483–505.

[9] J. R. B. COCKETT and P. J. W. HOFSTRA, *Introduction to Turing categories*. **Annals of Pure and Applied Logic**, vol. 156 (2008), pp. 183–209.

[10] H. B. CURRY, *Grundlagen der kombinatorischen Logik*. **American Journal of Mathematics**, vol. 52 (1930), pp. 509–536, 789–834.

[11] Y. L. ERSHOV, *Theorie der Numerierungen I*. **Zeitschrift für Mathematische Logik und Grundlagen der Mathematik**, vol. 19 (1973), pp. 289–388.

[12] ———, *Theorie der Numerierungen II*. **Zeitschrift für Mathematische Logik und Grundlagen der Mathematik**, vol. 21 (1975), pp. 473–584.

[13] S. Feferman, *A language and axioms for explicit mathematics*, **Algebra and Logic** (J. N. Crossley, editor), Springer, Berlin, 1975, pp. 87–139.

[14] R. M. Friedberg, *Three theorems on recursive enumeration*. **Journal of Symbolic Logic**, vol. 23 (1958), pp. 309–316.

[15] M. Hyland, *Some reasons for generalising domain theory*. **Mathematical Structures in Computer Science**, vol. 20 (2010), no. 2, pp. 239–265.

[16] S. C. Kleene and R. E. Vesley, **The Foundations of Intuitionistic Mathematics**, North-Holland, Amsterdam, 1965.

[17] G. Kreisel, *Some reasons for generalizing recursion theory*, **Logic Colloquium '69** (R. O. Gandy and C. E. M. Yates, editors), North-Holland, Amsterdam, 1971, pp. 139–198.

[18] M. Kummer, *Recursive enumeration without repetition revisited*, **Recursion Theory Week** (K. Ambos-Spies, G. H. Müller, G. E. Sacks, editors), Lecture Notes in Mathematics, vol. 1432, Springer-Verlag, Berlin, 1990, pp. 255–276.

[19] J. Longley, **Realizability toposes and language semantics**, Ph.D. thesis, Edinburgh University, 1995.

[20] J. Longley and D. Normann, **Higher-Order Computability**, Springer, Berlin, 2015.

[21] P. Odifreddi, **Classical Recursion Theory, vol. 1**, Studies in Logic and the Foundations of Mathematics, vol. 125, North-Holland, Amsterdam, 1989.

[22] H. Rogers, Jr., **Theory of Recursive Functions and Effective Computability**, McGraw Hill, New York, 1967.

[23] G. E. Sacks, **Higher Recursion Theory**, Springer-Verlag, Berlin, 1990.

[24] M. Schönfinkel, *Über die Bausteine der mathematischen Logik*. **Mathematische Annalen**, vol. 92 (1924), pp. 305–316.

[25] V. Selivanov, *Index sets of quotient objects of the Post numeration*. **Algebra i Logika**, vol. 27 (1988), no. 3, pp. 343–358 (English translation 1989).

[26] R. I. Soare, **Recursively Enumerable Sets and Degrees**, Springer-Verlag, Berlin, 1987.

[27] S. A. Terwijn, *Generalizations of the recursion theorem*. **Journal of Symbolic Logic**, vol. 83 (2018), no. 4, pp. 1683–1690.

[28] A. S. Troelstra and D. van Dalen, **Constructivism in Mathematics, vol. II**, Studies in Logic and the Foundations of Mathematics, vol. 123, North-Holland, Amsterdam, 1988.

[29] J. van Oosten, *A general form of relative recursion*. **Notre Dame Journal of Formal Logic**, vol. 47 (2006), no. 3, pp. 311–318.

[30] ———, **Realizability: An Introduction to Its Categorical Side**, Studies in Logic and the Foundations of Mathematics, vol. 152, Elsevier, Amsterdam, 2008.

[31] J. van Oosten and N. Voorneveld, *Extensions of Scott's graph model and Kleene's second algebra*. **Indagationes Mathematicae**, vol. 29 (2018), pp. 5–22.

[32] C. E. M. Yates, *Review of* [17]. **Journal of Symbolic Logic**, vol. 40 (1975), pp. 230–232.

DEPARTMENT OF MATHEMATICS
 RADBOUD UNIVERSITY NIJMEGEN
  P.O. BOX 9010, 6500 GL NIJMEGEN, THE NETHERLANDS
 E-mail: terwijn@math.ru.nl