Mathematical Structures in Computer Science (2024), **34**, pp. 455–466
doi:10.1017/S0960129524000252

PAPER

The complexity of completions in partial combinatory algebra

Sebastiaan Terwijn

Department of Mathematics, Radboud University Nijmegen, Nijmegen, GL, The Netherlands
Email: terwijn@math.ru.nl

(Received 11 September 2023; revised 15 May 2024; accepted 13 August 2024; first published online 23 September 2024)

Abstract

We discuss the complexity of completions of partial combinatory algebras, in particular, of Kleene's first model. Various completions of this model exist in the literature, but all of them have high complexity. We show that although there are no computable completions, there exist completions of low Turing degree. We use this construction to relate completions of Kleene's first model to complete extensions of PA. We also discuss the complexity of pcas defined from nonstandard models of PA.

Keywords: Partial combinatory algebra; Turing degrees; Peano arithmetic

1. Introduction

Partial combinatory algebra (pca) generalizes the setting of classical combinatory algebra (ca) to structures with a partial application operator. The first entry in the literature is Feferman (1975), which is surprisingly late, some fifty years after the invention of combinatory algebra and the closely related lambda calculus, although the concept of a pca existed before that (see Section 5). Apart from this connection with lambda calculus, pcas have played a notable part in constructive mathematics. At the end of Section 2 below, we list a number of key examples of pcas, and say something about their role in various settings.

Since the application operator in pcas is partial, they can often be naturally represented as c.e. structures in the sense of Selivanov (2003) and Khoussainov (2018). The computable structure theory of pcas as partial c.e. structures was recently studied by Fokina and Terwijn (2024). Since pcas can be seen as abstract models of computation, it is only natural to consider their complexity as algebraic structures from the viewpoint of computability theory. At least for countable pcas, there is a straightforward definition of their complexity in terms of the complexity of a presentation, as in computable model theory. Below we formulate this using numberings (Definition 3.1). Some of the complexity of pcas was studied earlier in Shafer and Terwijn (2021) and Golov and Terwijn (2023). In the current paper, we focus on the complexity of completions, and in particular of completions of what is called Kleene's first model $\mathcal{K}_1$, with application defined in terms of partial computable functions on the natural numbers.

A completion of a pca is a total pca (i.e. a combinatory algebra in the classical sense, in which applications are always defined) in which the pca can be embedded. Not every pca has a completion, as was first proved in Klop (1982). On the other hand, Kleene's $\mathcal{K}_1$ *does* have a completion. This follows from the sufficient condition for completability given in Bethke et al. (1996). This yields a certain term model $T(\omega)/{\sim}$ as a completion of $\mathcal{K}_1$. In Section 5 below, we will discuss how

Scott's graph model, which is another important example of a pca, can also be seen as a (weak) completion of $\mathcal{K}_1$. We note, however, that these completions of $\mathcal{K}_1$ have high complexity, which brings up the question of what the optimal complexity of such a completion could be. Although no computable completions of $\mathcal{K}_1$ exist (cf. Theorem 5.2 and the remarks following it), we show that there exist completions of $\mathcal{K}_1$ of low Turing degree (Theorem 5.3). Such completions are close to computable in the sense that the complexity of their halting problem is the same as the standard halting problem.

All this suggests a connection with complete extensions of Peano arithmetic, for which a similar story exists. Note, however, that we are talking here about pcas, that is, the models of a theory, rather than the theory itself. Nevertheless, in Section 6, we show that indeed there is a connection. Finally, in Section 7, we discuss the complexity of pcas resulting from nonstandard models of PA.

Our notation from computability theory is mostly standard. For unexplained notions, we refer to Odifreddi (1989) or Soare (1987). In particular, ω denotes the natural numbers, and φ_e the e-th partial computable (p.c.) function. For any set A, A' denotes the Turing jump of A, and in particular $\emptyset'$ denotes the halting problem.

2. Partial Combinatory Algebras

To make the paper self-contained, we briefly review the basic definitions from partial combinatory algebra. Our presentation follows van Oosten (2008).

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 usually 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, so 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_1t_2. If $t(x_1, \ldots, x_n)$ is a term with variables x_i, and $a_1, \ldots, a_n \in \mathcal{A}$, then $t(a_1, \ldots, a_n)$ 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_1t_2 to be defined, it is required that both t_1 and t_2 are defined.

Definition 2.1. *A pas $\mathcal{A}$ is called combinatory complete if for any term $t(x_1, \ldots, x_n, x)$, $n \geqslant 0$, with free variables among $x_1, \ldots, x_n, x$, there exists $b \in \mathcal{A}$ such that for all $a_1, \ldots, a_n, a \in \mathcal{A}$,*

(i) $ba_1 \cdots a_n\downarrow$,

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

A pas $\mathcal{A}$ is a pca if it is combinatory complete. A ca is a pca for which the application operator is total.

Combinatory completeness of pcas can be characterized by the existence of combinators k and s, just as in classical ca and lambda calculus. In van Oosten (2008), it is stated that the following theorem is "essentially due to Feferman (1975)."

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

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

In the following, we will always assume that our pcas are *nontrivial*, that is, have more than one element. This automatically implies that they are infinite and have $k \neq s$.

https://doi.org/10.1017/S0960129524000252 Published online by Cambridge University Press

The prime example of a pca is Kleene's first model $\mathcal{K}_1$ that was already mentioned in the introduction. This is a model defined on the natural numbers, with application

$$n \cdot m = \varphi_n(m).$$

Thus $\mathcal{K}_1$ models the setting of classical computability theory. We can also relativize this to an arbitrary oracle X, thus obtaining the relativized pca $\mathcal{K}_1^X$.

Kleene's second model $\mathcal{K}_2$, from the book Kleene and Vesley (1965), is a pca defined on Baire space ω^ω. Application $\alpha \cdot \beta$ in this model can be informally described as applying the continuous functional with code α to the real β. The original coding of $\mathcal{K}_2$ is a bit cumbersome, but it is essentially equivalent to

$$\alpha \cdot \beta = \Phi_{\alpha(0)}^{\alpha \oplus \beta},$$

where Φ_e is the e-th Turing functional, and the application is understood to be defined if the RHS is total. This coding, used in Shafer and Terwijn (2021), is easier to work with. See the appendix of Golov and Terwijn (2023) for a proof (and precise statement) of the equivalence with the original coding.

An interesting variant of $\mathcal{K}_2$, called the van Oosten model, is obtained by extending the domain to include partial functions, cf. van Oosten (1999). $\mathcal{K}_2$ is uncountable, but restricting attention to computable sequences gives a countable pca $\mathcal{K}_2^{\mathrm{eff}}$. Similarly, restricting to X-computable sequences gives a pca $\mathcal{K}_2^X$ for every X. In Golov and Terwijn (2023), the relations between these and other pcas are studied using embeddings.

Many other examples of pcas can be found in the literature. For example, pcas have been extensively used in constructive mathematics, see Beeson (1985), Troelstra and van Dalen (1988). In particular, they have been used as a basis for models of constructive set theory, as in McCarty (1986), Rathjen (2006), and Frittaion and Rathjen (2021). Pcas are also pivotal in the categorical treatment of realizability, cf. van Oosten (2008, Chapter 2) where they serve as a basis for realizability toposes. In particular, Hyland's famous effective topos is the realizability topos of $\mathcal{K}_1$. For a categorical characterization of pcas, see Cockett and Hofstra (2008) (extending early work of Longo and Moggi 1984); for a discussion of pcas in the context of oracles see Kihara (2022).

A pca that has been particularly important in connection with ca and lambda calculus is Scott's graph model (Scott 1975). This pca is a model of the lambda calculus (see Barendregt 1984), and it is also closely related to the enumeration degrees in computability theory, cf. Odifreddi (1999). We will discuss this model in Section 5, where we also explain how the restriction of this model to the c.e. sets can be seen as a completion of $\mathcal{K}_1$.

3. Effective Presentations of Pcas

Below, we will call a ca $\mathcal{A}$ Y-computable if $\mathcal{A}$ has a representation such that the application $\cdot$ in $\mathcal{A}$ is Y-computable. We will also require that equality on $\mathcal{A}$ is Y-decidable. The following definition (similar to notions used in Golov and Terwijn 2023) makes this precise, using numberings to represent $\mathcal{A}$. Recall that a numbering is a surjective function $\gamma : \omega \to \mathcal{A}$. We think of $n \in \omega$ as a code for $\gamma(n) \in \mathcal{A}$.

Definition 3.1. *Let $\mathcal{A}$ be a pca and $Y \subseteq \omega$. We call $\mathcal{A}$ partial Y-computable if there exist a numbering $\gamma : \omega \to \mathcal{A}$ and a partial Y-computable function ψ such that for all n and m, $\gamma(n) \cdot \gamma(m)\downarrow$ in $\mathcal{A}$ if and only if $\psi(n, m)\downarrow$, and*

$$\gamma(n) \cdot \gamma(m) = \gamma(\psi(n, m)). \tag{1}$$

We also require that equality on $\mathcal{A}$ is Y-decidable, meaning that the set $\{(n, m) \mid \gamma(n) = \gamma(m)\}$ is Y-computable. If $\mathcal{A}$ is total, i.e., a ca, then ψ is total, and we simply call $\mathcal{A}$ Y-computable. (This is consistent with Definition 3.2 below.)

https://doi.org/10.1017/S0960129524000252 Published online by Cambridge University Press

Notice that the numbering γ in Definition 3.1 is not required to be computable in any way. Also, nontrivial combinatorial algebras are never computable, cf. Barendregt (1984, 5.1.15).

Definition 3.1 focuses on representing application in a pca as a p.c. function. For the record, we also mention another way to define effective representations.

Definition 3.2. *We call a pca $\mathcal{A}$ Y-c.e. if there exist a numbering $\gamma : \omega \to \mathcal{A}$ such that the set*

$$\big\{(n, m, k) \mid \gamma(n) \cdot \gamma(m)\downarrow = \gamma(k)\big\} \tag{2}$$

is Y-c.e. Again we also require that equality on $\mathcal{A}$ is Y-decidable, meaning that the set $\{(n, m) \mid \gamma(n) = \gamma(m)\}$ is Y-computable. We call $\mathcal{A}$ Y-computable if the set (2) is Y-computable.[1]

As an example, note that $\mathcal{K}_1$ is p.c. in the sense that $a \cdot b$ is a p.c. function on ω, and that $\mathcal{K}_1$ is c.e. in the sense that $a \cdot b\downarrow = c$ is a c.e. relation. We note here that the two definitions are equivalent:

Proposition 3.3. *A pca is partial Y-computable if and only if it is Y-c.e.*

Proof. We always have c.e. implies p.c.: Given n, m, search for k such that (n, m, k) is in the set (2), and define $\psi(n, m)$ to be the least k found. (Note that γ need not be injective, so there may be multiple such k.) Then Equation (1) holds for $\psi(n, m)$.

The converse direction uses the condition that equality on $\mathcal{A}$ is decidable. Given $\mathcal{A}$ p.c. and ψ satisfying (1), enumerate (n, m, k) if $\psi(n, m)\downarrow$ and $\gamma(k) = \gamma(\psi(n, m))$. This gives an enumeration of Equation (2). □

Note that the above definitions are in the spirit of the c.e. structures in Selivanov (2003), where they are called *positive* structures. These are defined as structures in which the predicates are c.e., and the functions are computable. The latter makes sense for total functions, but in the case of pcas, we are dealing with a partial application operator, in which case it is natural to have this as a c.e. function.

For pcas on ω (i.e. with $\gamma : \omega \to \mathcal{A}$ the identity), we have that equality on $\mathcal{A}$ is decidable, so the two notions of pca are equivalent. This is the type of pca that was used in Fokina and Terwijn (2024). Without the condition that equality on $\mathcal{A}$ is decidable (or c.e.), it is not clear that the two definitions are equivalent in general, though we do not know of an example of a pca that is p.c. but not c.e.

Finally, in the case of a Y-computable pca or ca $\mathcal{A}$ (which is what we will mostly use below), the requirement that equality on $\mathcal{A}$ is Y-decidable actually *follows* from the fact that Equation (2) is Y-computable, namely let n be such that $\gamma(n)$ is the identity (which exists in any pca).

4. Embeddings and Isomorphisms

There are at least three notions of embedding for pcas, depending on what structure is required to be preserved. For example, the choice of the combinators k and s from Theorem 2.2 can be regarded as part of the structure or not. For instance, Zoethout (2022, p. 33) does not consider k and s to be part of the structure of a pca. We have the following notions of embedding of pcas:

- Only preserve applications. This notion was studied in Bethke (1988), Asperti and Ciabattoni (1997), Shafer and Terwijn (2021), and Golov and Terwijn (2023).
- Besides applications, also preserve k and s, for a particular choice of these combinators. This stronger notion was studied in Bethke et al. (1996, 1999).

- There is an even weaker notion of embedding, using the notion of applicative morphism, that was introduced in Longley (1994), see also Longley and Normann (2015). Applicative morphisms do not have to preserve applications; instead, there have to be terms in the codomain that simulate applications in the domain. This notion is useful in realizability theory, see van Oosten (2008).

Our primary interest here is the notion of embedding where k and s are not considered part of the signature, but we will also be using the stronger notion of embedding, especially when we talk about completions. To distinguish the two, we will refer to them as weak and strong embeddings. (In Golov and Terwijn 2023, weak embeddings were simply called embeddings.) To distinguish applications in different pcas, we also write $\mathcal{A} \models a \cdot b\downarrow$ if this application is defined in $\mathcal{A}$.

Definition 4.1. *For given pcas $\mathcal{A}$ and $\mathcal{B}$, an injection $f : \mathcal{A} \to \mathcal{B}$ is a weak embedding if for all $a, b \in \mathcal{A}$,*

$$\mathcal{A} \models ab\downarrow \implies \mathcal{B} \models f(a)f(b)\downarrow = f(ab). \tag{3}$$

If $\mathcal{A}$ embeds into $\mathcal{B}$, in this way we write $\mathcal{A} \hookrightarrow \mathcal{B}$. If in addition to (3), for a specific choice of combinators k and s of $\mathcal{A}$, $f(k)$ and $f(s)$ serve as combinators for $\mathcal{B}$, we call f a strong embedding.

A (total) ca $\mathcal{B}$ is called a weak completion of $\mathcal{A}$ if there exists a weak embedding $\mathcal{A} \hookrightarrow \mathcal{B}$. If the embedding is strong, we call $\mathcal{B}$ a strong completion.

Two pcas $\mathcal{A}$ and $\mathcal{B}$ are isomorphic, denoted by $\mathcal{A} \cong \mathcal{B}$, if there exists a bijection $f : \mathcal{A} \to \mathcal{B}$ such that for all $a, b \in \mathcal{A}$, $ab\downarrow$ if and only if $f(a)f(b)\downarrow$, and in this case

$$f(a) \cdot f(b) = f(ab).$$

Besides the term completion, in the literature also the term extension is used. Bethke et al. (1999, Definition 1.5) call a pca $\mathcal{B}$ an *extension* of a pca $\mathcal{A}$ if $\mathcal{A} \subseteq \mathcal{B}$, the application $\cdot_{\mathcal{A}}$ in $\mathcal{A}$ is the restriction of application $\cdot_{\mathcal{B}}$ in $\mathcal{B}$ to the domain of $\cdot_{\mathcal{A}}$, and $\mathcal{B}$ and $\mathcal{A}$ both have the *same* combinators k and s as in Theorem 2.2.

Now suppose that $f : \mathcal{A} \hookrightarrow \mathcal{B}$ is a strong embedding. Then $f(\mathcal{A}) \subseteq \mathcal{B}$ is an extension in the above sense, where both $f(\mathcal{A})$ and $\mathcal{B}$ have combinators $f(k)$ and $f(s)$. Note that $\mathcal{A} \cong f(\mathcal{A})$ if we define application in $f(\mathcal{A})$ by $f(\mathcal{A}) \models f(a) \cdot f(b)\downarrow = f(c)$ if and only if $\mathcal{A} \models a \cdot b\downarrow = c$. So we see that total extensions and completions amount to the same thing, provided that in both cases we have to specify whether to also fix s and k or not.

In Terwijn (2023), it is shown that weak and strong embeddability and completions are different: There exists a pca that is weakly completable, but not strongly completable.[2]

5. Complexity of Completions of $\mathcal{K}_1$

It was an important open question in the 1970s whether every pca has a strong completion. The question was raised by Barendregt, Mitschke, and Scott, and discussed at a meeting in Swansea in 1974, cf. Bethke et al. (1999). (Note that this predates Feferman's paper Feferman 1975.) A negative answer was obtained by Klop (1982), see also Bethke et al. (1999). Other examples of incompletable pcas can be found in Bethke (1987) and Bethke and Klop (1996).

In contrast to these examples, $\mathcal{K}_1$ does have strong completions. This follows from the criterion given in Bethke et al. (1996) about the existence of unique head-normal forms, which is satisfied in $\mathcal{K}_1$. The completion of $\mathcal{K}_1$ resulting from this is a certain term model $T(\omega)/\!\sim$. On the face of it, the equivalence relation $\sim$ is not computable, since it is essentially equivalence of terms in $\mathcal{K}_1$. That indeed it cannot be computable follows from Theorem 5.2, and also from the fact that computable combinatorial algebras do not exist.

https://doi.org/10.1017/S0960129524000252 Published online by Cambridge University Press

We now discuss how another famous pca can be seen as a completion of $\mathcal{K}_1$. Scott's graph model $\mathcal{G}$ is a pca defined on the power set $\mathcal{P}(\omega)$, with application defined by

$$X \cdot Y = \big\{x \mid \exists u(\langle x, u\rangle \in X \wedge D_u \subseteq Y)\big\}.$$

Here D_u as always denotes the finite set with canonical code u, and $\langle\cdot,\cdot\rangle$ denotes an effective pairing function. $\mathcal{E}$ is defined as the restriction of $\mathcal{G}$ to the c.e. sets. That $\mathcal{G}$ and $\mathcal{E}$ are (total) cas is implicit in Scott (1975). Note the close connection with enumeration reducibility (cf. Odifreddi 1999, XIV): For all sets Y and Z, $Z \leqslant_e Y$ is equivalent with $X \cdot Y = Z$ for some c.e. set X.

In Golov and Terwijn (2023, Corollary 7.5, it was shown that $\mathcal{K}_1 \hookrightarrow \mathcal{E}$, so that we can see $\mathcal{E}$ as a weak completion of $\mathcal{K}_1$. Note that equality on $\mathcal{E}$ is equality of c.e. sets, which is Π^0_2-complete when we represent c.e. sets by their indices.[3] So this is more complicated than equality in the term model $T(\omega)/{\sim}$.

We can see $\mathcal{E}$ as a combination of $\mathcal{K}_1$ and $\mathcal{K}_2$. Indeed we have

$$\mathcal{K}_1 \hookrightarrow \mathcal{E} \hookrightarrow \mathcal{K}_2$$

(the latter by Golov and Terwijn 2023, Corollary 6.2, so that we can view $\mathcal{E}$ as a kind of middle ground between Kleene's models. This combination famously gives a model of the λ-calculus, as shown in Scott (1975), see Odifreddi (1999, XIV.4).

Below, we use that for an embedding $f : \mathcal{K}_1 \hookrightarrow \mathcal{A}$ of $\mathcal{K}_1$ into a pca $\mathcal{A}$, it suffices to know the value of f on finitely many elements. This observation was also used in Golov and Terwijn (2023, Theorem 4.1), and it can be used to bypass the fact that embeddings such as f do not have to be computable. Below we give a somewhat simpler version of this trick, using the following lemma.

Lemma 5.1. *There exist elements $t_n \in \mathcal{K}_1$, $n \geqslant 1$, such that for all n and m,*

$$t_n \cdot m = \begin{cases} n & \text{if } m = 0 \\ t_{n+1} & \text{if } m > 0. \end{cases}$$

Proof. By the recursion theorem, let $d \in \omega$ be a code such that

$$\varphi_d(n, m) = \begin{cases} n & \text{if } m = 0 \\ S^1_1(d, n+1) & \text{if } m > 0. \end{cases}$$

Here S^1_1 is the primitive recursive function from the S-m-n-theorem. Define $t_n = S^1_1(d, n)$. W.l.o.g. we may assume $t_n > 0$ for all n. Then $\varphi_{t_n}(m) = \varphi_{S^1_1(d,n)}(m) = \varphi_d(n, m) = t_{n+1}$ for $m > 0$ and equal to n otherwise. □

In Golov and Terwijn (2023, Corollary 4.2, it was proved that if $\mathcal{K}^X_1 \hookrightarrow \mathcal{A}$ is a weak embedding of $\mathcal{K}^X_1$ into a pca $\mathcal{A}$ with Y-c.e. inequality, then $X \leqslant_T Y$. We obtain a stronger conclusion when we assume that $\mathcal{A}$ is total and Y-computable.

Theorem 5.2. *Suppose $\mathcal{A}$ is a Y-computable combinatorial algebra, and that $f : \mathcal{K}^X_1 \hookrightarrow \mathcal{A}$ is a weak embedding. Then $X <_T Y$.*

Proof. The successor function S in $\mathcal{K}_1$ satisfies $S^n(0) = n$, but we need an element t such that the n-fold application $t \cdot \ldots \cdot t \cdot 0$ equals n, with the convention that application associates to the left, not to the right. Let $t = t_1$ be as in Lemma 5.1. Then for the n-fold application, we have $t \cdot \ldots \cdot t \cdot 0 = t_n \cdot 0 = n$ for every $n > 0$. Since f is an embedding, we obtain from this

$$f(n) = f(t \cdot \ldots \cdot t \cdot 0) = f(t) \cdot \ldots \cdot f(t) \cdot f(0), \tag{4}$$

https://doi.org/10.1017/S0960129524000252 Published online by Cambridge University Press

with the applications repeated n times. So we see that the image of f is completely determined by $f(t)$ and $f(0)$.

To show that $Y \not\leqslant_T X$, let A, B be a X-computably inseparable pair of X-c.e. sets, and let e be a code such that for all x,

$$\varphi_e^X(x) = \begin{cases} 0 & \text{if } x \in A \\ 1 & \text{if } x \in B \\ \uparrow & \text{otherwise.} \end{cases}$$

Then we have in particular that

$$\mathcal{K}_1^X \models e \cdot x{\downarrow} = 0 \Longrightarrow \mathcal{A} \models f(e) \cdot f(x) = f(0),$$
$$\mathcal{K}_1^X \models e \cdot x{\downarrow} = 1 \Longrightarrow \mathcal{A} \models f(e) \cdot f(x) = f(1).$$

Since $\mathcal{A}$ is total, for every x the application $f(e) \cdot f(x)$ is always defined in $\mathcal{A}$, and by Equation (4), it is equal to a term containing only $f(t)$, $f(0)$, and application. Because $\mathcal{A}$ is Y-computable, we can compute a code of $f(e) \cdot f(x)$ effectively from x. (All we need is e, and codes of $f(t)$ and $f(0)$, all of which are fixed.) Furthermore, since the definition of Y-computable pca entails that equality on $\mathcal{A}$ is Y-decidable, we can decide with Y whether $f(e) \cdot f(x)$ is equal to $f(0)$ or $f(1)$ or not. It follows that the set $C = \{x \mid \mathcal{A} \models f(e) \cdot f(x) = f(0)\}$ is Y-computable and separates A and B, and since A and B have no X-computable separation Y is not X-computable.

That $X \leqslant_T Y$ can be shown using a very similar argument. Instead of φ_e^X above, use the characteristic function $\varphi_d^X(x)$ which is 0 if $x \in X$ and 1 if $x \notin X$. Then the rest of the argument above, replacing e with d, shows that X is Y-computable. So we have $Y \not\leqslant_T X$ and $X \leqslant_T Y$, hence $X <_T Y$. □

From Theorem 5.2, we see that, in particular, $\mathcal{K}_1$ does not have a computable weak completion, which also follows from the fact that combinatorial algebras are never computable, see Barendregt (1984, 5.1.15). We now show that this is optimal, namely that there exist completions of low Turing degree. (Recall that Y is low if $Y' \leqslant_T \emptyset'$).

Theorem 5.3. *There exists a strong completion $\mathcal{A}$ of $\mathcal{K}_1$ of low Turing degree, that is, $\mathcal{A}$ is Y-computable such that Y is low.*

Proof. The outline of the proof is as follows. We first define a first-order base theory Cmpl such that each model of Cmpl gives rise to a strong completion of $\mathcal{K}_1$. The theory Cmpl will be consistent because we already know that $\mathcal{K}_1$ has strong completions. We then use standard recursion theory to obtain a complete and consistent extension of Cmpl of low degree. This does not immediately give a completion of $\mathcal{K}_1$ of low degree, but we use a model-theoretic argument to obtain a completion of the desired complexity.

The language of Cmpl is two-sorted,[4] with a predicate $N(x)$ intended to range over natural numbers, and a predicate $A(x)$ intended to range over a pca $\mathcal{A}$ that is a completion of $\mathcal{K}_1$. Furthermore, the language has a function symbol f with the intended meaning that $f : \mathcal{K}_1 \to \mathcal{A}$ is a strong embedding. The language for the sort N is the same as the language of arithmetic, and for this sort, we take the axioms of PA. The language of the sort A is that of pcas, with one function symbol $\cdot$ for application in $\mathcal{A}$. Since $\cdot$ will be required to be total we add it as a function symbol, rather than as a relation symbol, which would have been more appropriate for a partial operation. By arithmetization, we may assume that expressions of the form $\varphi_a(b){\downarrow} = c$ are directly expressible for the sort N for all standard numbers $a, b, c \in \omega$, where we represent a number $n \in \omega$ by the term $S^n(0)$.

So as axioms of Cmpl, we have the following:

- The axioms of PA for the sort N (i.e. all axioms relative to N).
- Axioms expressing that f is an embedding from $\mathcal{K}_1$ to $\mathcal{A}$:
  - $\forall a \in N(f(a) \in A)$.
  - $\forall a, b \in N(f(a) = f(b) \to a = b)$.
  - For all $a, b, c \in \omega$, we have an axiom
  $$\mathcal{K}_1 \models a \cdot b{\downarrow} = c \implies \mathcal{A} \models f(a) \cdot f(b) = f(c). \tag{5}$$
  Note that the LHS can be expressed for the sort N using the language of arithmetic, using terms $S^n(0)$ to express the natural number n, and the RHS can be expressed for the sort A.
- To ensure that f is a strong embedding, we fix standard combinators s and k in $\mathcal{K}_1$ satisfying the axioms of Theorem 2.2. Note that these can be expressed for the sort N. Also, note that s and k are just standard numbers, so we do not need to add them to the signature. Next, we add axioms expressing that $f(s)$ and $f(k)$ also satisfy the axioms of Theorem 2.2, but now for the sort A. The existence of these combinators $f(s)$ and $f(k)$ automatically ensures that $\mathcal{A}$ forms a pca. The fact that $\mathcal{A}$ should be total is handled by the fact that application is a function symbol in the language, so no explicit axiom is needed for this.

Taken together, the axioms of Cmpl express that f is a strong embedding from $\mathcal{K}_1$ to $\mathcal{A}$. Every model M of Cmpl gives a strong completion of $\mathcal{K}_1$ as follows. Denote by $M \restriction N$ and $M \restriction A$ the part of M restricted to the sorts N and A. Then $M \restriction N$ is a model of PA and $\mathcal{A} = M \restriction A$ is a pca. Furthermore, the restriction of f^M to the standard numbers $n \in \omega$ is an injection of ω into $\mathcal{A}$, which, by Equation (5), is an embedding of $\mathcal{K}_1$, which is strong because $f(s)$ and $f(k)$ satisfy the axioms of Theorem 2.2. The values of f^M on possible nonstandard elements of $M \restriction N$ are irrelevant.

Since Cmpl is a computable axiomatization, by a result of Shoenfield (cf. Cenzer 1999, Theorem 6.1), the set of complete and consistent extensions of it can be represented as a Π^0_1-class, that is, there is a computable tree $T \subseteq 2^{<\omega}$ such that the set of infinite paths $[T]$ consists of all complete and consistent extensions of Cmpl. (We encode sentences by natural numbers, so that paths in 2^ω correspond to sets of sentences.) Note that the theory Cmpl is consistent because we know by Bethke et al. (1996) that there exists a strong completion of $\mathcal{K}_1$. In particular, the tree T is infinite, and $[T]$ is nonempty. By the Low Basis Theorem (Jockusch and Soare 1972), T has a path of low Turing degree, which gives us a complete and consistent extension X of Cmpl of low degree.

Since X is consistent, it has a model M by the completeness theorem, and by the remarks above M defines a strong completion of $\mathcal{K}_1$, namely $M \restriction A$. However, there is no guarantee that this completion is X-computable. But we do not need all of $M \restriction A$; it suffices to consider the smaller pca $\mathcal{A}$ consisting of all terms built from $f(n)$ for standard numbers $n \in \omega$ (represented as terms $S^n(0)$), and application. Note that $\mathcal{A}$ is a pca because of the presence of $f(s)$ and $f(k)$, and $\mathcal{A}$ is total since for u and v of the given form, $u \cdot v$ is again of this form. Not every element of $\mathcal{A}$ is of the form $f(n)$, for example, it has terms $f(a) \cdot f(b)$ such that $\varphi_a(b){\uparrow}$. The pca $\mathcal{A}$ is a sub-pca of $M \restriction A$ in the sense of Shafer and Terwijn (2021). To finish the proof of the theorem, we note that $\mathcal{A}$ is X-computable. Namely, given to terms u and v of the form above, we can simply compute their application as the term $u \cdot v$. Equality of terms in $\mathcal{A}$ is X-decidable because the theory X is complete and thus contains all equalities $u = v$ and $u \neq v$ of such terms. So the sub-pca $\mathcal{A}$ of $M \restriction A$ is total and X-computable, and hence of low degree since X is low. □

6. Complete Extensions of PA

Following modern terminology, we call a Turing degree a PA degree if it is the degree of a complete extension of Peano arithmetic (cf. Downey and Hirschfeldt 2010, p. 84). We will simply call a set PA-complete if it has PA degree.

https://doi.org/10.1017/S0960129524000252 Published online by Cambridge University Press

In this section, we show that every (strong or weak) completion of $\mathcal{K}_1$ computes a PA degree, and vice versa. Since there exist PA-complete sets of low degree (Downey and Hirschfeldt 2010, p. 87), Theorem 5.3 follows from the statement of this equivalence; however, this does not make the proof of Theorem 5.3 superfluous, since the tree T from its proof is used in the proof of the equivalence.

Proposition 6.1. *Every PA-complete set computes a strong completion of $\mathcal{K}_1$.*

Proof. By results of Scott and Solovay (cf. Odifreddi 1989, V.5.36), a set Y is PA-complete if and only if it can compute an element of every nonempty Π^0_1-class. In particular, Y can compute an element of $[T]$ for the computable tree T from the proof of Theorem 5.3. By the rest of the proof of Theorem 5.3, this implies that Y computes a strong completion of $\mathcal{K}_1$, namely the term model defined at the end of the proof. □

The following result strengthens Theorem 5.2.

Theorem 6.2. *Suppose $\mathcal{A}$ is a Y-computable combinatorial algebra, and that $f : \mathcal{K}_1 \hookrightarrow \mathcal{A}$ is a weak embedding. Then Y is PA-complete.*

Proof. According to Jockusch and Soare (1972), a set is PA-complete if and only if it can compute a separation of an effectively inseparable pair of c.e. sets. (See also Downey and Hirschfeldt 2010, p. 86.) Now let A, B be a pair of effectively inseparable c.e. sets, for example, we can take the provable and refutable sentences of PA (Odifreddi 1989, p. 513). Then the proof of Theorem 5.2 shows that Y computes a separation of A and B, and hence Y is PA-complete. □

Putting Proposition 6.1 and Theorem 6.2 together, we obtain the following characterization:

Corollary 6.3. *The following are equivalent for any set A:*

(i) *A computes a weak completion of $\mathcal{K}_1$,*
(ii) *A computes a strong completion of $\mathcal{K}_1$,*
(iii) *A is PA-complete.*

In the case of PA degrees, more is known, namely that they are closed upwards. We do not know whether the degrees of (weak or strong) completions of $\mathcal{K}_1$ are also upwards closed.

7. Nonstandard Models of PA

As mentioned in Beeson (1985, VI.2.5) and van Oosten (2008), every model M of Peano Arithmetic PA defines a pca on M, with application defined by

$$a \cdot b\downarrow = c \text{ if } M \models \varphi_a(b)\downarrow = c \tag{6}$$

for all $a, b, c \in M$. By restricting Equation (6) to standard numbers $a, b, c \in \omega$, we obtain a pca on ω, which we will call $\mathcal{K}_1(M)$. Note that $\mathcal{K}_1(M)$ is just Kleene's first model $\mathcal{K}_1$ "inside M." It is a pca because we can pick combinators $k, s \in \mathcal{K}_1$ as in Theorem 2.2 such that PA proves that they have the required properties.

Note that for $a, b, c \in \omega$ we have that $a \cdot b\downarrow = c$ in $\mathcal{K}_1$ if and only if $\exists y (T(a, b, y) \wedge U(y) = c)$, where T and U are the primitive recursive predicate and function from Kleene's normal form theorem (cf. Odifreddi 1989). In a nonstandard model M, this y can be nonstandard, so that *more*

https://doi.org/10.1017/S0960129524000252 Published online by Cambridge University Press

computations converge than in reality. In particular, in general, we have

$$\mathcal{K}_1 \models a \cdot b\downarrow = c \Longrightarrow \mathcal{K}_1(M) \models a \cdot b\downarrow = c, \tag{7}$$

but not conversely. For example, consider a model M of PA + ¬con(PA), where con(PA) expresses the consistency of PA. Such models exist by Gödel's second incompleteness theorem. If we consider the p.c. function φ_a that on input b searches for a proof of an inconsistency in PA, then the computation $a \cdot b$ will converge in $\mathcal{K}_1(M)$, but not in $\mathcal{K}_1$ (assuming PA is consistent).

Note that by Equation (7), we have an embedding $\mathcal{K}_1 \hookrightarrow \mathcal{K}_1(M)$ for every model M of PA. This is in fact a strong embedding, as the same combinators s and k can be used in $\mathcal{K}_1(M)$.

Since PA proves that certain p.c. functions are nontotal, any model of PA has nontotal p.c. functions. This remains true if we restrict to standard numbers, that is, there are standard numbers $e, n \in \omega$ such that PA proves that $\varphi_e(n)$ never halts. In particular, we see that $\mathcal{K}_1(M)$ is never a total pca (i.e. a ca). Therefore, we have:

Proposition 7.1. *$\mathcal{K}_1(M)$ is never a weak completion of $\mathcal{K}_1$.*

By Tennenbaum's theorem (cf. Boolos and Jeffrey 1974), there are no computable nonstandard models of PA. More precisely, there are no nonstandard models in which + is computable. (This is an extension of Tennenbaum's theorem due to Kreisel.) It follows that there are also no nonstandard models that are c.e., because + is a total operation, and in a c.e. model, it would actually be computable, contradicting Kreisel's result. So it would seem that the pcas $\mathcal{K}_1(M)$ for nonstandard models M cannot be used for the problems about c.e. pcas discussed in computable structure theory (see Fokina and Terwijn 2024). However, for models M_0 and M_1 of PA, we do *not* have in general that

$$M_0 \not\cong M_1 \Longrightarrow \mathcal{K}_1(M_0) \not\cong \mathcal{K}_1(M_1).$$

To see this, let $M_0 = \omega$ be the standard model, and let M_1 be a nonstandard model that has the same first-order theory as ω (which exists by the compactness theorem). Then $M_0 \not\cong M_1$, but $\mathcal{K}_1(M_0) = \mathcal{K}_1(M_1) = \mathcal{K}_1$.

In particular, we see that it is possible that $\mathcal{K}_1(M)$ is c.e. (in the sense of Definition 3.2) for a nonstandard model M of PA. This prompts the following question:

Question 7.2. What are the possible c.e. degrees for such $\mathcal{K}_1(M)$aaaa can $\mathcal{K}_1(M)$ be c.e. but not equal to $\mathcal{K}_1$aaaa □

In the following, we note that though $\mathcal{K}_1(M)$ is always noncomputable, it can have low degree (if we do not require that it is also c.e.).

Proposition 7.3. *$\mathcal{K}_1(M)$ is always noncomputable.*

Proof. Consider the computably inseparable pair of sets

$$A = \{x \in \omega \mid \varphi_x(x)\downarrow = 0\}$$
$$B = \{x \in \omega \mid \varphi_x(x)\downarrow = 1\}.$$

Now consider the set $C = \{x \in \omega \mid \mathcal{K}_1(M) \models x \cdot x\downarrow = 0\}$. C is computable from $\mathcal{K}_1(M)$, and it follows from Equation (7) that it separates A and B, from which it follows immediately that $\mathcal{K}_1(M)$ cannot be computable. □

Proposition 7.4. *$\mathcal{K}_1(M)$ can have low Turing degree.*

https://doi.org/10.1017/S0960129524000252 Published online by Cambridge University Press

Proof. We have to show that there exists a model M and a low set Y such that $\mathcal{K}_1(M)$ is Y-computable (in the sense of Definition 3.2), that is, such that the set

$$Z = \{(a, b, c) \mid \mathcal{K}_1(M) \models a \cdot b{\downarrow} = c\}$$

is Y-computable. According to Jockusch and Soare, there exists a complete and consistent extension X of PA of low Turing degree. Let M be a model with theory X. We can identify the set Z with the set of sentences $a \cdot b{\downarrow} = c$ that hold in M. Since Z is then just a subset of X consisting of sentences of a specific form, there is a computable set R such that $Z = R \cap X$, and we have $(R \cap X)' \leqslant_T X' \leqslant_T \emptyset'$ so that Z is low. □

Competing interests. The author declares none.

Notes

1 Note that this definition of computable pca is different from the definition of decidable pca in van Oosten (2008, Definition 1.3.7), which refers to the decidability of equality inside the pca, using an element of the pca.

2 The argument runs as follows: First, $\mathcal{K}_2$ has strong completions. Second, the counterexample from Bethke et al. (1999) of a pca without strong completions can be weakly embedded into $\mathcal{K}_2$. Hence this weak embedding cannot be made strong.

3 In the discussion of $\mathcal{E}$ as a model of the λ-calculus, Odifreddi (1999, p. 858) also defines an application on ω by $e \cdot x =$ index of $W_e \cdot W_x$. Odifreddi says that this choice of application is "equivalent" to $\mathcal{E}$. However, this application on ω does not give a pca, as equality on ω is decidable, so this would contradict that $\mathcal{K}_1$ does not have a computable weak completion (cf. Theorem 5.2). So to obtain a pca, we have to divide out by equivalence of c.e.-indices, which gives precisely $\mathcal{E}$. Also note that the model of the λ-calculus really uses c.e. sets, not indices.

4 For more about multi-sorted languages and models, see Monk (1976, p. 483 ff). It is well-known that languages with finitely many sorts, as in our case, reduce to ordinary first-order logic by using predicates for the various sorts, as we do here directly. There is no need to keep the sorts N and A disjoint, so we have what is called a *lax* setting. When writing axioms for a sort, instead of writing $\forall a(N(a) \to \varphi)$, we also simply write $\forall a \in N.\, \varphi$.

References

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

Barendregt, H. P. (1984). *The Lambda Calculus*, Studies in Logic and the Foundations of Mathematics, vol. **103**, 2nd edn., Amsterdam, North-Holland.

Beeson, M. J. (1985). *Foundations of Constructive Mathematics*. Springer: Berlin, Heidelberg.

Bethke, I. (1987). On the existence of extensional partial combinatory algebras. *Journal of Symbolic Logic* **52** (3) 819–833.

Bethke, I. (1988). Notes on Partial Combinatory Algebras. Phd thesis, Universiteit van Amsterdam.

Bethke, I. and Klop, J. W. (1996). Collapsing partial combinatory algebras. In: Dowek, G., Heering, J., Meinke, K. and Möller, B.(eds.), vol. **1074**, 57–73, *Springer Lecture Notes in Computer Science*

Bethke, I., Klop, J. W. and de Vrijer, R. (1996). Completing partial combinatory algebras with unique head-normal forms. In: *Proceedings of 11th Annual IEEE Symposium on Logic in Computer Science*, IEEE Computer Society Press, 448–454.

Bethke, I., Klop, J. W. and de Vrijer, R. (1999). Extending partial combinatory algebras. *Mathematical Structures in Computer Science* **9** 483–505.

Boolos, G. S. and Jeffrey, R. C. (1974). *Computability and Logic*, New York, Cambridge University Press.

Cenzer, D. (1999). Π^0_1-classes in computability theory. In: *Handbook of Computability Theory*, Griffor, E. R. (ed.), Amsterdam, North-Holland, 37–85.

Cockett, J. R. B. and Hofstra, P. J. W. (2008). Introduction to Turing categories. *Annals of Pure and Applied Logic* **156** 183–209.

Downey, R. G. and Hirschfeldt, D. R. (2010). *Algorithmic Randomness and Complexity*. Springer: New York.

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

Fokina, E. and Terwijn, S. A. (2024). *Computable Structure Theory of Partial Combinatory Algebras.* In: Computability in Europe (CiE 2024), Lecture Notes in Computer Science 14773, Springer: Cham, 265–276.

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

Golov, A. and Terwijn, S. A. (2023). Embeddings between partial combinatory algebras. *Notre Dame Journal of Formal Logic* **64** (1) 129–158.

Jockusch, C. G., Jr. and Soare, R. I. (1972). Π^0_1-classes and degrees of theories. *Transactions of the American Mathematical Society* **173** 35–56.

Khoussainov, B. (2018). A journey to computably enumerable structures, tutorial lectures. In: *Computability in Europe 2018*, Manea, F. et al. (ed.), LNCS vol. **10936**, Springer, 1–19.

Kihara, T. (2022). Rethinking the notion of oracle, preprint, arXiv.

Kleene, S. C. and Vesley, R. E. (1965). *The Foundations of Intuitionistic Mathematics*, North-Holland: Amsterdam.

Klop, J. W. (1982). Extending partial combinatory algebras. *Bulletin of the European Association for Theoretical Computer Science* **16** 472–482.

Longley, J. (1994). Realizability Toposes and Language Semantics, PhD thesis, University of Edinburgh.

Longley, J. and Normann, D. (2015). *Higher-Order Computability*, Springer: Berlin, Heidelberg.

Longo, G. and Moggi, E. (1984). Gödel numberings, principal morphisms, combinatory algebras. In: *Mathematical Foundations of Computer Science*, Chytil, C. and Koubek, K. (ed.), LNCS, vol. **176**, Springer Verlag.

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

Monk, J. D. (1976). *Mathematical Logic*, Springer: New York.

Odifreddi, P. G. (1989). Classical recursion theory. *Studies in Logic and the Foundations of Mathematics*, vol. **125**, North–Holland.

Odifreddi, P. G. (1999). *Classical Recursion Theory*, Studies in Logic and the Foundations of Mathematics, vol. **143**, Amsterdam, North-Holland.

Rathjen, M. (2006). Realizability for constructive Zermelo-Fraenkel set theory. In: *Logic Colloquium'03*, 282–314.

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

Selivanov, V. (2003). Positive structures. In: *Computability and Models*, Cooper, S. B. and Goncharov, S. S. (ed.), Kluwer, 321–350.

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

Soare, R. I. (1987). *Recursively Enumerable Sets and Degrees*, Springer: Berlin, Heidelberg.

Terwijn, S. A. (2023). Completions of Kleene's second model, arXiv.

Troelstra, A. S. and van Dalen, D. (1988). *Constructivism in Mathematics*, Studies in Logic and the Foundations of Mathematics, vol. **123**, North-Holland.

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

van Oosten, J. (2008). *Realizability: An Introduction to Its Categorical Side*, Studies in Logic and the Foundations of Mathematics, vol. **152**, Elsevier.

Zoethout, J. (2022). Computability Models and Realizability Toposes. Phd thesis, Utrecht University.

Cite this article: Terwijn S (2024). The complexity of completions in partial combinatory algebra. *Mathematical Structures in Computer Science* **34**, 455–466. https://doi.org/10.1017/S0960129524000252