Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

Completing partial combinatory algebras with unique head-normal forms

I. Bethke, J.W. Klop and R. de Vrijer

Computer Science/Department of Software Technology

CS-R9544 1995

Report CS-R9544
ISSN 0169-118X

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

Completing Partial Combinatory Algebras with Unique Head-Normal Forms

Inge Bethke[1,2,4] Jan Willem Klop[1,3,5] Roel de Vrijer[3,6]

[1] *CWI*
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

[2] *University of Utrecht, Department of Philosophy*
P.O. Box 80126, 3508 TC Utrecht, The Netherlands

[3] *Vrije Universiteit, Department of Mathematics and Computer Science*
de Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

[4]inge@cwi.nl [5]jwk@cwi.nl [6]rdv@cs.vu.nl

Abstract

In this note, we prove that having unique head-normal forms is a sufficient condition on partial combinatory algebras to be completable. As application, we show that the pca of strongly normalizing CL-terms as well as the pca of natural numbers with partial recursive function application can be extended to total combinatory algebras.

AMS Subject Classification (1991): 08A55, 03B40, 16S15.
CR Subject Classification (1991): F.4.1.
Keywords & Phrases: Partial algebras, combinatory logic, λ-calculus, term rewriting.

1. Introduction

A *partial combinatory algebra (pca)* is a structure $\mathfrak{A} = <A, s, k, \cdot>$ where A is a set, $\cdot$ is a partial binary operation (application) on A, and k, s are two elements of A such that

1. $\forall a, a' \in A \;\; (k \cdot a) \cdot a' = a$,
2. $\forall a, a' \in A \;\; (s \cdot a) \cdot a' \downarrow$,
3. $\forall a, a', a'' \in A \;\; ((s \cdot a) \cdot a') \cdot a'' = \begin{cases} (a \cdot a'') \cdot (a' \cdot a'') & \text{if } (a \cdot a'') \cdot (a' \cdot a'') \downarrow, \\ \text{undefined} & \text{otherwise,} \end{cases}$
4. $k \neq s$.

Here $M\downarrow$ means the expression M is defined, and $M = N$ means both expressions are defined and equal. It is common to omit $\cdot$ and associate unparenthesized expressions to the left. In working with expressions that may or may not be defined, it is useful to write $M \simeq N$ to mean that if either M or N is defined, then both are defined and equal. These notational conventions allow us to replace clause 3 by

$$\forall a, a', a'' \in A \quad saa'a'' \simeq aa''(a'a'').$$

Total pca's, where application is a total operation on the carrier set, will be called *ca's*, and nontotal pca's, where application is not defined everywhere, will be called *nca's*.

In this paper, we are interested in the possibility of embedding a given nca $\mathfrak{A}$ into a ca. We shall call such an embedding a *completion* of $\mathfrak{A}$. More precisely: let $\mathfrak{A} = <A, s, k, \cdot>$ and $\mathfrak{B} = <B, s', k', \cdot'>$ be pca's.

1. A *homomorphism* of $\mathfrak{A}$ into $\mathfrak{B}$ is a mapping $\phi : A \rightarrow B$ such that
   (a) $\phi(s) = s'$, $\phi(k) = k'$, and
   (b) if $a \cdot a' \downarrow$ then $\phi(a \cdot a') = \phi(a) \cdot' \phi(a')$ for all $a, a' \in A$.

   If ϕ is injective, then ϕ is an *embedding*.

2. ϕ is a *completion* of $\mathfrak{A}$ if ϕ is an embedding of $\mathfrak{A}$ into some ca $\mathfrak{B}$.

We say that $\mathfrak{A}$ *has a completion* or is *completable* if there exists some completion of $\mathfrak{A}$. Not every nca is completable. Examples of these incompletable nca's can be found in e.g. [Klo82], [Bet87] and [BK95].

Given a pca $\mathfrak{A} = <A, s, k, \cdot>$, we call elements of A of the forms s, k, ka, sa, saa' *head-normal forms (hnf)*. Each of the five types of hnf is called *dissimilar* from the other four. Moreover, we say that a pca $\mathfrak{A}$ *has unique hnf's* if

1. no two dissimilar hnf's can be equal in $\mathfrak{A}$;
2. Barendregt's axiom (cf. [Bar75]) holds in $\mathfrak{A}$:

$$(BA) \qquad sa_0a_1 = sa_2a_3 \Rightarrow a_0 = a_2 \ \wedge \ a_1 = a_3.$$

In [Klo82], the second author advanced the theorem that having unique hnf's is a sufficient condition on pca's to be completable. In this paper, we shall prove this theorem in detail.

2. How to complete pca's with unique hnf's

In order to prove the theorem, we employ a free-algebra construction induced by a term rewrite system. The construction is based on fundamental definitions and notions of term rewrite systems. Extensive surveys of term rewriting can be found in [Klo92] and [DJ90].

Let $\mathfrak{A} = <A, s, k, \cdot>$ be a pca. The *term rewrite system over* $\mathfrak{A}$, $\mathcal{T}(\mathfrak{A})$, consists of

1. $T(A \cup V)$, the set of terms built from A, a countably infinite set V of variables, and a binary function symbol $*$ (written infix), and
2. $\mathcal{R}$, the set of the following rewrite rules:
   (a) $a * a' \rightarrow aa'$ provided $aa' \downarrow$,

(b) $k * x * y \rightarrow x$,
(c) $ka * x \rightarrow a$,
(d) $s * x * y * z \rightarrow (x * z) * (y * z)$,
(e) $sa * x * y \rightarrow (a * y) * (x * y)$, and
(f) $saa' * x \rightarrow (a * x) * (a' * x)$.

with $a, a' \in A$. Here we employ the convention of association to the left.

Identity of terms is denoted $\equiv$. A *substitution* σ is a mapping from V to $T(A \cup V)$. Substitutions are extended to homomorphisms from $T(A \cup V)$ to $T(A \cup V)$. The set of *contexts* over $T(A \cup V)$, $\mathcal{C}$, is defined as follows.

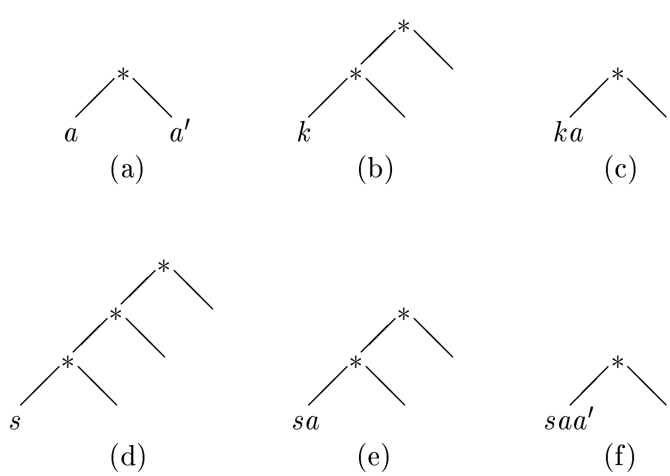
1. $\Box \in \mathcal{C}$, and
2. if $C \in \mathcal{C}$ and $t \in T(A \cup V)$, then $t * C \in \mathcal{C}$ and $C * t \in \mathcal{C}$.

If C is a context, then $C[t]$ denotes the term obtained from C by replacing $\Box$ by t. The *rewrite relation* $\rightarrow$ associated with $\mathcal{T}(\mathfrak{A})$ is defined as follows: $t \rightarrow t'$ if there exists a rewrite rule $l \rightarrow r$ in $\mathcal{R}$, a substitution σ and a context C such that $t \equiv C[\sigma(l)]$ and $t' \equiv C[\sigma(r)]$. The transitive-reflexive closure of $\rightarrow$ is denoted by $\twoheadrightarrow$. If $t \twoheadrightarrow t'$ we say that t *reduces* to t'. We write $t \leftarrow t'$ if $t' \rightarrow t$; likewise for $t \twoheadleftarrow t'$. The equivalence relation generated by $\rightarrow$ is called *convertibility* and written as $\sim$. $\mathcal{T}(\mathfrak{A})$ is *confluent* if

$$\forall t, t' \in T(A \cup V) \; (\; t \sim t' \Rightarrow \exists t'' \in T(A \cup V) \; t \twoheadrightarrow t'' \twoheadleftarrow t' \;).$$

In order to prove that $\mathcal{T}(\mathfrak{A})$ is confluent, we shall subdivide $\mathcal{T}(\mathfrak{A})$ into the two separate rewrite systems $\mathcal{T}_1(\mathfrak{A}) = (T(A \cup V), \mathcal{R}_1)$ and $\mathcal{T}_2(\mathfrak{A}) = (T(A \cup V), \mathcal{R}_2)$ where $\mathcal{R}_1$ consists of the rewrite schema (a), and $\mathcal{R}_2$ consists of the remaining schemas (b)-(f).

A *pattern* of a rewrite rule $t \rightarrow t'$ is the part of t's construction tree that does not contain any variables. Observe that $\mathcal{T}_1(\mathfrak{A})$ and $\mathcal{T}_2(\mathfrak{A})$ have the following patterns:



A term rewrite system is *orthogonal* if it is *left-linear*, i.e. no variable occurs twice or more in the left-hand term of any rule, and *non-ambiguous*, i.e. has the property that in no term patterns can overlap. Orthogonal term rewrite systems have the confluence property as well as various other desirable properties concerned with reduction strategies.

$\mathcal{T}_1(\mathfrak{A})$ is clearly orthogonal. As for $\mathcal{T}_2(\mathfrak{A})$, we can only state with certainty that it is left-linear; overlap of patterns, however, can occur if and only if

1. two dissimilar hnf's are equal in $\mathfrak{A}$, or
2. there are $a_0 \neq a_1$, $a_2 \neq a_3$ such that $ka_0 = ka_1$, $sa_0 = sa_1$, or $sa_0a_2 = sa_1a_3$.

It follows that $\mathcal{T}_2(\mathfrak{A})$ is orthogonal too provided $\mathfrak{A}$ has unique hnf's. For, if $\mathfrak{A}$ has unique hnf's, then

1. no two dissimilar hnf's are equal in $\mathfrak{A}$,
2. (a) if $ka_0 = ka_1$, then $a_0 = ka_0k = ka_1k = a_1$,
   (b) if $sa_0 = sa_1$, then $sa_0k = sa_1k$ and hence $a_0 = a_1$ by (BA),
   (c) if $sa_0a_2 = sa_1a_3$, then $a_0 = a_1$ and $a_2 = a_3$ again by (BA).

Proposition 2.1. *Let $\mathfrak{A} =< A, s, k, \cdot >$ be a pca with unique hnf's. Then both $\mathcal{T}_1(\mathfrak{A})$ and $\mathcal{T}_2(\mathfrak{A})$ are orthogonal and, a fortiori, confluent.* □

We shall use confluence of its subsystems to prove that $\mathcal{T}(\mathfrak{A})$ is confluent. For this, we invoke a proposition that is sometimes referred to as the *Lemma of Hindley-Rosen* ([Hin64], [Bar84]): For $i = 1, 2$, let us write $\rightarrow_i$ for the rewrite relation associated with $\mathcal{T}_i(\mathfrak{A})$. The reflexive closure of $\rightarrow_i$ is denoted by $\overset{\equiv}{\rightarrow}_i$, its transitive-reflexive closure by $\twoheadrightarrow_i$. Moreover, we say that $\twoheadrightarrow_1$ and $\twoheadrightarrow_2$ *commute*, if

$$\forall t, t', t'' \in T(A \cup V)\, \exists t''' \in T(A \cup V)\ (\, t \twoheadrightarrow_1 t' \ \wedge\ t \twoheadrightarrow_2 t'' \Rightarrow t' \twoheadrightarrow_2 t''' \ \wedge\ t'' \twoheadrightarrow_1 t''' \,).$$

Now, given the confluence of $\mathcal{T}_1(\mathfrak{A})$ and $\mathcal{T}_2(\mathfrak{A})$, the Lemma of Hindley-Rosen states that $\mathcal{T}(\mathfrak{A})$ is confluent, if $\twoheadrightarrow_1$ and $\twoheadrightarrow_2$ commute. However, as observed in [Hin64], commutativity of $\twoheadrightarrow_1$ and $\twoheadrightarrow_2$ already follows if the following diagram commutes:

$$\begin{array}{ccc} t & \longrightarrow_1 & t' \\ {\scriptstyle 2}\big\downarrow & & \big\downarrow{\scriptstyle 2}\ \equiv \\ t'' & \twoheadrightarrow_1 & t''' \end{array}$$

We shall use this strengthened version of the Lemma of Hindley-Rosen in the proposition below.

PROPOSITION 2.2. *Let $\mathfrak{A} = <A, s, k, \cdot>$ be a pca with unique hnf's. Then $\mathcal{T}(\mathfrak{A})$ is confluent.*

PROOF. We have to check all possible diagrams of the sort depicted above. To this end, let us call a substitution instance of the left-hand term of rewrite rule (i) (i $\in \{\mathrm{a,b,c,d,e,f}\}$) an *(i)-redex*. Moreover, let us write $t \to_i t'$ for the reduction of t to t' obtained by an application of rewrite rule (i); likewise for $t \twoheadrightarrow_i t'$ and $t \overset{\equiv}{\to}_i t'$. Observe that, if the left-hand upper expression t contains an (a)-redex $a * a'$ disjoint from an (i)-redex l (i $\in \{\mathrm{b,c,d,e,f}\}$), in the upper horizontal reduction step the (a)-redex is contracted, and in the left vertical reduction step the (i)-redex is contracted, then the diagram commutes trivially:

$$\begin{array}{ccc} t \equiv C[l, a * a'] & \longrightarrow_a & t' \equiv C[l, aa'] \\ {\scriptstyle i}\downarrow & & {\scriptstyle i}\downarrow \\ t'' \equiv C[l', a * a'] & \longrightarrow_a & t''' \equiv C[l', aa'] \end{array}$$

here C is a context containing two holes $\Box$, and the common reduct t''' is obtained by contracting the (i)-redex l in t' and the (a)-redex $a * a'$ in t''. It remains to consider the cases where the redexes are not disjoint, i.e. where one redex is a subexpression of the other. In these cases we can actually forget about the surrounding context C and can focus on the positions of the redexes relative to each other. Since an (i)-redex can never be a proper subexpression of an (a)-redex, it remains to consider the cases where the (a)-redex is a subexpression of the (i)-redex for i $\in \{\mathrm{b,c,d,e,f}\}$. There are in fact 14 such cases which we have arranged in groups depending on i. To obtain a more compact notation, we abbreviate expressions of the form $C[t]$ to C_t and let a, b, c range over elements of A. We believe that the diagrams are self-explanatory and do not require any further comment.

Case i=b:

$$\begin{array}{ccc} k * a * t & \longrightarrow_a & ka * t \\ {\scriptstyle b}\downarrow & & {\scriptstyle c}\downarrow \\ a & \twoheadrightarrow_a & a \end{array} \qquad \begin{array}{ccc} k * C_{a*b} * t & \longrightarrow_a & k * C_{ab} * t \\ {\scriptstyle b}\downarrow & & {\scriptstyle b}\downarrow \\ C_{a*b} & \longrightarrow_a & C_{ab} \end{array}$$

$$\begin{array}{ccc} k * t * C_{a*b} & \longrightarrow_a & k * t * C_{ab} \\ {\scriptstyle b}\downarrow & & {\scriptstyle b}\downarrow \\ t & \twoheadrightarrow_a & t \end{array}$$

Case i=c:

$$\begin{array}{ccc} ka * b & \longrightarrow_a & kab = a \\ {\scriptstyle c}\downarrow & & {\scriptstyle c}\downarrow \equiv \\ a & \twoheadrightarrow_a & a \end{array} \qquad \begin{array}{ccc} ka * C_{b*c} & \longrightarrow_a & ka * C_{bc} \\ {\scriptstyle c}\downarrow & & {\scriptstyle c}\downarrow \\ a & \twoheadrightarrow_a & a \end{array}$$

Case i=d:

$$\begin{array}{ccc} s * a * t * t' & \longrightarrow_a & sa * t * t' \\ {\scriptstyle d}\downarrow & & {\scriptstyle e}\downarrow \\ a * t' * (t * t') & \twoheadrightarrow_a & a * t' * (t * t') \end{array} \qquad \begin{array}{ccc} s * C_{a*b} * t * t' & \longrightarrow_a & s * C_{ab} * t * t' \\ {\scriptstyle d}\downarrow & & {\scriptstyle d}\downarrow \\ C_{a*b} * t' * (t * t') & \longrightarrow_a & C_{ab} * t' * (t * t') \end{array}$$

$$\begin{array}{ccc} s * t * C_{a*b} * t' & \longrightarrow_a & s * t * C_{ab} * t' \\ {\scriptstyle d}\downarrow & & {\scriptstyle d}\downarrow \\ t * t' * (C_{a*b} * t') & \longrightarrow_a & t * t' * (C_{ab} * t') \end{array} \qquad \begin{array}{ccc} s * t * t' * C_{a*b} & \longrightarrow_a & s * t * t' * C_{ab} \\ {\scriptstyle d}\downarrow & & {\scriptstyle d}\downarrow \\ t * C_{a*b} * (t' * C_{a*b}) & \twoheadrightarrow_a & t * C_{ab} * (t' * C_{ab}) \end{array}$$

Case i=e:

$$\begin{array}{ccc} sa * b * t & \longrightarrow_a & sab * t \\ {\scriptstyle e}\downarrow & & {\scriptstyle f}\downarrow \\ a * t * (b * t) & \twoheadrightarrow_a & a * t * (b * t) \end{array} \qquad \begin{array}{ccc} sa * C_{b*c} * t & \longrightarrow_a & sa * C_{bc} * t \\ {\scriptstyle e}\downarrow & & {\scriptstyle e}\downarrow \\ a * t * (C_{b*c} * t) & \longrightarrow_a & a * t * (C_{bc} * t) \end{array}$$

$$\begin{array}{ccc} sa * t * C_{b*c} & \longrightarrow_a & sa * t * C_{bc} \\ {\scriptstyle e}\downarrow & & {\scriptstyle e}\downarrow \\ a * C_{b*c} * (t * C_{b*c}) & \twoheadrightarrow_a & a * C_{bc} * (t * C_{bc}) \end{array}$$

Case i=f:

$$\begin{array}{ccc} sab * c & \longrightarrow_a & sabc = ac(bc) \\ {\scriptstyle f}\downarrow & & {\scriptstyle f}\downarrow \equiv \\ a * c * (b * c) & \twoheadrightarrow_a & ac(bc) \end{array} \qquad \begin{array}{ccc} sab * C_{c*d} & \longrightarrow_a & sab * C_{cd} \\ {\scriptstyle f}\downarrow & & {\scriptstyle f}\downarrow \\ a * C_{c*d} * (b * C_{c*d}) & \twoheadrightarrow_a & a * C_{cd} * (b * C_{cd}) \end{array}$$

□

If $\mathcal{T}(\mathfrak{A})$ is confluent, then the following quotient construction provides a completion of $\mathfrak{A}$.

Definition 2.3. Let $\mathfrak{A} =<A, s, k, \cdot>$ be a pca.

1. Let $T(A) \subset T(A \cup V)$ be the set of all closed terms, i.e. terms without any variable. We form the quotient

   $$\Gamma(\mathfrak{A}) =<T(A)/\sim, [s], [k], \cdot>$$

   by taking the collection

   $$T(A)/\sim = \{[t] \mid t \in T(A)\}$$

   of equivalence classes

   $$[t] = \{t' \in T(A) \mid t \sim t'\}$$

   equipped with the total application operation

   $$[t] \cdot [t'] = [t * t'].$$

2. Define $\gamma_{\mathfrak{A}} : A \to T(A)/\sim$ by

   $$\gamma_{\mathfrak{A}}(a) = [a]$$

   for all $a \in A$.

Theorem 2.4. *Let $\mathfrak{A} =<A, s, k, \cdot>$ be a pca with unique hnf's. Then $\gamma_{\mathfrak{A}}$ is a completion of $\mathfrak{A}$.*

Proof. We have to prove that

1. $\Gamma(\mathfrak{A})$ is a ca, and
2. $\gamma_{\mathfrak{A}}$ is an embedding of $\mathfrak{A}$ into $\Gamma(\mathfrak{A})$.

First observe that

$$(\dagger) \quad [a] = [a'] \Rightarrow a = a'$$

for all $a, a' \in A$: For, if $[a] = [a']$, then $a \sim a'$. Hence a and a' have a common reduct, since $\mathcal{T}(\mathfrak{A})$ is confluent. Therefore, as a, a' cannot be reduced any further, $a = a'$.
(1.) Since $s \neq k$, $[s] \neq [k]$. Hence $\Gamma(\mathfrak{A})$ meets the fourth condition on pca's. It clearly meets the second condition, since application is total. Satisfaction of condition 1. and 2. follows from the rewrite rules (b) and (d), respectively.
(2.) Clearly, $\gamma_{\mathfrak{A}}$ preserves the constants. For preservation of application, let $a, a' \in A$ be such that $aa' \downarrow$. Then $a * a' \sim aa'$ by rewrite rule (a). Thus

$$\gamma_{\mathfrak{A}}(aa') = [aa'] = [a * a'] = [a] \cdot [a'] = \gamma_{\mathfrak{A}}(a) \cdot \gamma_{\mathfrak{A}}(a').$$

So $\gamma_{\mathfrak{A}}$ is an homomorphism and is injective by ($\dagger$). □

In the next section, we shall discuss two examples.

3. Examples

For nca's, we can reduce the property of having unique head-normal forms to a more handsome set of five axioms.

Proposition 3.1. *Let $\mathfrak{A} =<A, s, k, \cdot>$ be a nca. $\mathfrak{A}$ has unique head-normal forms if and only if $\mathfrak{A}$ satisfies Barendregt's axiom as well as the following four axioms: for all $a, a', a'' \in A$,*

1. $s \neq saa'$,
2. $k \neq saa'$,
3. $sa \neq sa'a''$, *and*
4. $ka \neq sa'a''$.

Proof. The if-part is obvious. For the only-if-part, we have to show that the remaining dissimilar hnf's are unequal in $\mathfrak{A}$. That is, we have to show that for all $a, a' \in A$,

5. $s \neq ka$: Suppose $s = ka$. Then $ss = kas = a = kak = sk$. Hence $s = k$ by (BA). Contradiction.
6. $s \neq sa$: Suppose $s = sa$ and pick $a', a'' \in A$ such that $a'a''$ is undefined. Then $sa'a'' = saa'a'' \simeq aa''(a'a'')$. Hence $a'a''$ is defined. Contradiction.

7. $k \neq ka$: Suppose $k = ka$. Then $ks = kas = a = kak = kk$. Hence $s = kss = kks = k$. Contradiction.

8. $k \neq sa$: Suppose $k = sa$ and pick $a', a'' \in A$ such that $a'a''$ is undefined. Then $a' = ka'a'' = saa'a'' \simeq aa''(a'a'')$. Hence $a'a''$ is defined. Contradiction.

9. $ka \neq sa'$: Suppose $ka = sa'$. Then $sa'k = kak = a = kas = sa's$. Hence $s = k$ by (BA). Contradiction.

□

We shall use this shorter characterization in the two examples to follow.

EXAMPLE 3.2. The term rewrite system CL of combinatory logic consists of

1. $T(\{S, K\} \cup V)$, the set of terms built from the two constants S, K, a countably infinite set V of variables, and a binary application operator $\cdot$ which we do not write, and

2. the following two rewrite rules:

   (a) $Sxyz \to xz(yz)$

   (b) $Kxy \to x$.

The rewrite relation associated with CL is defined as usually, i.e. as in the case of $\mathcal{T}(\mathfrak{A})$. As is well-known, CL is confluent.

A term of the form $SLMN$ or KLM is a *redex*. A term not containing such redexes is a *normal form (nf)* and has a nf if it reduces to one. A *reduction* of L is a sequence of terms $L \equiv L_1 \to L_2 \to L_3 \to \cdots$. Reductions may be infinite. If every reduction of L terminates eventually (in a normal form), then L is said to be *strongly normalizing*. We let SN be the set of all closed, strongly normalizing CL-terms.

Closed, strongly normalizing terms modulo convertibility form a pca in the following way (cf. also [BK95]): We let

$$\mathfrak{A}_{SN} =< \{[M]_{SN} \mid M \in SN\}, [S]_{SN}, [K]_{SN}, \cdot >$$

where

$$[M]_{SN} = \{N \in SN \mid M \sim N\}$$

and

$$[M]_{SN} \cdot [N]_{SN} = \begin{cases} [MN]_{SN} & \text{if } MN \in SN, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

The structure $\mathfrak{A}_{SN}$ is in fact an nca: $\omega \equiv S(SKK)(SKK) \in SN$ and hence $[\omega]_{SN}$ exists in $\mathfrak{A}_{SN}$. However, $\omega\omega \notin SN$; $[\omega]_{SN} \cdot [\omega]_{SN}$ is therefore undefined.

We shall not prove in detail that $\mathfrak{A}_{SN}$ has unique head-normal forms, but merely consider Barendregt's axiom. The argument for the satisfaction of the remaining axioms is similar. Thus assume $[S]_{SN}[M]_{SN}[N]_{SN} = [S]_{SN}[M']_{SN}[N']_{SN}$, i.e. $SMN \sim SM'N'$. Since CL is confluent, it follows that SMN and $SM'N'$ have a common reduct, say L. Moreover, since neither SMN nor $SM'N'$ is a redex, L must be of the form $SM''N''$ with $M \twoheadrightarrow M'' \twoheadleftarrow M'$ and $N \twoheadrightarrow N'' \twoheadleftarrow N'$. So $M \sim M'$ and $N \sim N'$, i.e. $[M]_{SN} = [M']_{SN}$ and $[N]_{SN} = [N']_{SN}$.

The completability of $\mathfrak{A}_{SN}$ does not come as a surprise. In fact, the codomain of its canonical completion, $\Gamma(\mathfrak{A}_{SN})$, is isomorphic to the paradigmatic ca $\mathfrak{A}_{CL}$ obtained from CL by taking as carrier the set of all closed CL-terms modulo convertibility.

Example 3.3. As second example we consider the nca of natural numbers with partial recursive function application. More specifically, we define a nontotal application operation on the natural numbers $\mathbb{N}$ by

$$n \cdot m = \{n\}(m)$$

where $\{n\}$ is the partial recursive function with Gödel number n. It is not difficult to see that $<\mathbb{N}, \cdot>$ can be made into an nca by choosing appropriate Gödel numbers s and k. In fact, one can (effectively) generate infinitely many other indices which do the job. In what follows, we consider a particular nca on $<\mathbb{N}, \cdot>$ where the constants s and k are chosen in a way such that Theorem 2.4 applies.

If we fix a certain number of variables in a partial recursive function f, we still get a partial recursive function g of the remaining variables. Moreover, this can be done uniformly in the fixed variables. This is Kleene's famous S^m_n-Theorem (see also [Kle52]) which more precisely stated reads: Given $m, n \in \mathbb{N}$, there is a primitive recursive *injection* $S^m_n : \mathbb{N}^{m+1} \to \mathbb{N}$ such that

$$\{S^m_n(x, y_1, \ldots, y_m)\}(z_1, \ldots, z_n) \simeq \{x\}(y_1, \ldots, y_m, z_1, \ldots, z_n)$$

for all $x, y_1, \ldots, y_m, z_1, \ldots, z_n \in \mathbb{N}$.

We now fix pairwise distinct $n_1, n_2, n_3, n_4 \in \mathbb{N}$ such that for all $x, y, z \in \mathbb{N}$

1. $\{n_1\}(x, y, z) = S^2_1(x, y, z)$,
2. $\{n_2\}(x, y, z) = S^2_1(x, y, z)$,
3. $\{n_3\}(x, y, z) = y$, and
4. $\{n_4\}(x, y, z) \simeq \{\{x\}(z)\}(\{y\}(z))$,

and define

(k) $k = S^2_1(n_1, n_3, n_3)$, and

(s) $s = S^2_1(n_2, n_2, n_4)$.

Observe that k and s are chosen properly:

1. $$\begin{aligned} k \cdot n \cdot m &\simeq \{\{k\}(n)\}(m) \\ &\simeq \{\{S^2_1(n_1, n_3, n_3)\}(n)\}(m) \\ &\simeq \{\{n_1\}(n_3, n_3, n)\}(m) \\ &\simeq \{S^2_1(n_3, n_3, n)\}(m) \\ &\simeq \{n_3\}(n_3, n, m) = n, \end{aligned}$$

2. $$\begin{aligned} s \cdot n \cdot m &\simeq \{\{s\}(n)\}(m) \\ &\simeq \{\{S^2_1(n_2, n_2, n_4)\}(n)\}(m) \\ &\simeq \{\{n_2\}(n_2, n_4, n)\}(m) \\ &\simeq \{S^2_1(n_2, n_4, n)\}(m) \\ &\simeq \{n_2\}(n_4, n, m) = S^2_1(n_4, n, m) \qquad \text{and hence } s \cdot n \cdot m \downarrow, \end{aligned}$$

3. $$\begin{aligned} s \cdot n \cdot m \cdot o &\simeq \{S^2_1(n_4, n, m)\}(o) \qquad \text{by 2.} \\ &\simeq \{n_4\}(n, m, o) \\ &\simeq \{\{n\}(o)\}(\{m\}(o)) \simeq n \cdot o \cdot (m \cdot o), \end{aligned}$$

4. clearly $s \neq k$, since S^2_1 is injective.

So $\mathfrak{A}_{\mathbb{N}} = <\mathbb{N}, s, k, \cdot>$ is an nca. To prove that it has unique head-normal forms we invoke Proposition 3.1. That is, we have to prove

(BA): Suppose $s \cdot n \cdot m = s \cdot n' \cdot m'$. Then $S^2_1(n_4, n, m) = s \cdot n \cdot m = s \cdot n' \cdot m' = S^2_1(n_4, n', m')$ and hence $n = n'$ and $m = m'$.

3.1.1: $s = S^2_1(n_2, n_2, n_4) \neq S^2_1(n_4, n, n') = s \cdot n \cdot n'$ for all $n, n' \in \mathbb{N}$.

3.1.2: $k = S^2_1(n_1, n_3, n_3) \neq S^2_1(n_4, n, n') = s \cdot n \cdot n'$ for all $n, n' \in \mathbb{N}$.

3.1.3: $s \cdot n = \{S^2_1(n_2, n_2, n_4)\}(n) = \{n_2\}(n_2, n_4, n) = S^2_1(n_2, n_4, n) \neq S^2_1(n_4, n', n'') = s \cdot n' \cdot n''$ for all $n, n', n'' \in \mathbb{N}$.

3.1.4: $k \cdot n = \{S^2_1(n_1, n_3, n_3)\}(n) = \{n_1\}(n_3, n_3, n) = S^2_1(n_3, n_3, n) \neq S^2_1(n_4, n', n'') = s \cdot n' \cdot n''$ for all $n, n', n'' \in \mathbb{N}$.

So $\mathfrak{A}_{\mathbb{N}}$ has unique head-normal forms and is therefore completable by Theorem 2.4. It remains the question whether $\Gamma(\mathfrak{A}_{\mathbb{N}})$ is (isomorphic to) a well-known ca, or whether it is a latent model which deserves closer inspection.

REFERENCES

[Bar75] H.P. Barendregt. Normed uniformly reflexive structures. In C. Böhm, editor, *λ-Calculus and Computer Science Theory*, volume 37 of *Lecture Notes in Computer Science*, pages 272–286. Springer-Verlag, 1975.

[Bar84] H.P. Barendregt. *The Lambda Calculus, its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Company, revised edition, 1984. (Second printing 1985).

[Bet87] I. Bethke. On the existence of extensional partial combinatory algebras. *Journal of Symbolic Logic*, 52(3):819–833, 1987.

[BK95] I. Bethke and J.W. Klop. Collapsing partial combinatory algebras. Technical Report CS-R9520, CWI, 1995.

[DJ90] N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Formal Methods and Semantics, Handbook of Theoretical Computer Science, Volume B*, chapter 6, pages 243–320. MIT Press, 1990.

[Hin64] J.R. Hindley. *The Church-Rosser property and a result in combinatory logic*. PhD thesis, University of Newcastle-upon-Tyne, 1964.

[Kle52] S.C. Kleene. *Introduction to metamathematics*. North-Holland Publishing Company, Amsterdam, 1952.

[Klo82] J.W. Klop. Extending partial combinatory algebras. *Bulletin of the European Association for Theoretical Computer Science*, 16:472–482, 1982.

[Klo92] J.W. Klop. Term rewriting systems. In D. Gabbay S. Abramsky and T. Maibaum, editors, *Handbook of Logic in Computer Science, Volume II*. Oxford University Press, 1992.