The Combinatory Algebras $\mathcal{U}$ and $\mathcal{A}_{wb,\mathit{eff}}$ are not equivalent

Diplomarbeit
am Fachbereich Mathematik der
TECHNISCHEN UNIVERSITÄT DARMSTADT

in der Arbeitsgruppe
Logik und mathematische Grundlagen der Informatik
unter Betreuung von
Prof. Dr. Th. STREICHER

Karl Gernot Belger

Abstract

We will show that in the realisability model over the combinatory algebra $\mathcal{A}_{wb,\mathit{eff}}$ of effective well-bracketed strategies, due to Abramsky and Longley [AL99], the object $(N \to N_\perp) \to N_\perp$ is not projective.

On the other hand, in the realisability model presented by Marz, Rohr and Streicher in [MRS99], every object occurring as denotation of some PCF type is projective.

We conclude from this result that these two combinatory algebras (and hence the induced realisability models) are not equivalent.

Acknowledgements

The first person to thank here is my supervisor, Thomas Streicher. Without his endless patience and his many ideas this diploma thesis would not exist.

I also wish to thank the members of the research group *Arbeitsgruppe 14* for their constant support, especially Micha Marz and Alex Rohr for their proof reading and their many useful hints.

Last but not least I like to thank my parents who gave me moral and financial support throughout my studies.

Contents

1 Introduction

1.1 Background and Motivation

Most models for programming languages, in the sense of *denotational semantics* (see [Mos90] for an introduction), can be reconstructed as *realisability models*.

A realisability model is a category defined over a (partial) combinatory algebra which embodies some notion of 'low level computability'. See [Lon94] for a detailed introduction to realisability models.

One particular language, due to Scott [Sco69] and Plotkin [Plo77], is the well-known PCF. Defined to be a prototypical functional programming language, PCF is simply typed lambda calculus augmented by general recursion and basic arithmetic.

It was known from the outset that models for PCF with good properties exist (e.g. the Scott-Model). Nevertheless, it was long an open question whether there is a syntax-free model for PCF which is fully abstract. See [Ong95] for an survey article on the full abstraction problem for PCF.

In the last view years, essentially two different solutions to the full abstraction problem where found. On the one hand, the *game models*, independently found by Abramsky, Jagadeesan and Malacaria [AJM94], Hyland and Ong [HO95], as well as Nickau [Nic94]. On the other hand, the *relational model*, an approach using Kripke Logical Relations, found by O'Hearn and Riecke [OR95].

Both models were reconstructed as realisability models for PCF whose effective parts have an even stronger property than full abstraction, namely *universality*. Universality means that every morphism in the model is a denotation of some program.

Inspired by the category $\mathcal{SD}$ of sequential domains (the relational model due to M. Marz [Mar00]), Marz, Rohr and Streicher presented a combinatory algebra $\mathcal{U}$ in [MRS99].

Implementing ideas coming from the game models, Abramsky and Longley constructed the combinatory algebra of effective well-bracketed strategies, $\mathcal{A}_{wb,\mathit{eff}}$, in [AL99].

A natural question which arises now is whether these two realisability models are equivalent. This question will be the subject of the present diploma thesis. We will be able to give the negative answer that the two models are not equivalent.

1.2 Overview

In the next section, Section 2, we will introduce the notions of combinatory algebras, categories of assemblies and modest sets, dominances and divergences, as well as, briefly, PCF models.

In the following sections, two combinatory algebras, $\mathcal{U}$ and $\mathcal{A}_{wb,\mathit{eff}}$, will be introduced, both giving rise to universal models for PCF. In Section 3 we will sketch the construction of the combinatory algebra $\mathcal{U}$ and present the result that in the category of modest sets over $\mathcal{U}$ every object occurring as denotation of some PCF type is projective. In Section 4, we introduce the combinatory algebra of history-free strategies $\mathcal{A}$ and identify $\mathcal{A}_{wb,\mathit{eff}}$ as a subalgebra. We will then find good representations of the natural numbers object N and the object $N \to N_\perp$ in the category of modest sets over $\mathcal{A}_{wb,\mathit{eff}}$. Moreover, we will show a useful characterisation of well-bracketedness.

The main result of this diploma thesis will be that type $(N \to N_\perp) \to N_\perp$ is not projective in the category of modest sets over $\mathcal{A}_{wb,\mathit{eff}}$. This will be proven in the fifth section.

In the sixth and last section some hints for further work will be given.

1.3 Prerequisites and Notations

I assume that the reader is familiar with basic notions of category theory, recursion theory and lambda-calculus.

We write $f(x) \simeq g(x)$ for two partial functions $f, g\colon X \rightharpoonup Y$ and any $x \in X$, if they are 'Kleene equal', i.e. both sides are defined and equal or both sides diverge. We write $f(x)\downarrow$, if $f(x)$ is defined and $f(x)\uparrow$ otherwise.

I we define partial maps by

$$\begin{array}{rcl} f\colon X & \rightharpoonup & Y \\ x_1 & \mapsto & y_1 \\ \vdots & & \vdots \\ x_n & \mapsto & y_n \end{array}$$

it is understood that f is undefined on elements different from $x_1, \ldots, x_n$.

The set of natural numbers is denoted by $\mathbb{N}$. We write $\mathrm{Seq}(\mathbb{N})$ for the set of finite sequences of natural numbers.

For objects A and B in a category $\mathcal{C}$, we call A a *retract* of B (via an injection-projection pair (e, p)), and write $A \lhd B$, if there are morphisms $e\colon A \to B$ and $p\colon B \to A$ such that $p \circ e = \mathrm{id}_A$.

2 Preliminaries

In this section we introduce the notions of combinatory algebra, categories of assemblies and modest sets, dominances and divergences, as well as PCF models. Only a very condensed overview over these topics is given here to provide a minimal amount of self-containedness.

Most results are stated without proof and are taken from the thesis of John Longley [Lon94], who gives a detailed and thorough introduction to realisability models.

2.1 Combinatory Algebras

We work with the notion of total combinatory algebras instead of partial ones (which would be the more general approach) because all investigated combinatory algebras are total.

Definition 2.1 *A combinatory algebra* $(A, \cdot)$ *is a set* A *together with a binary operation* $\cdot\colon A \times A \to A$ *such that there exist* $k, s \in A$ *satisfying*

$$(k \cdot a) \cdot b = a \qquad \textit{and} \qquad \big((s \cdot a) \cdot b\big) \cdot c = (a \cdot c) \cdot (b \cdot c)$$

for all elements a, b, c *of* A.

We simply write $a\,b$ for $a{\cdot}b$ and employ the convention of left-associativity.

Very useful for the work with combinatory algebras is the combinatory completeness property. In order to introduce it we first need the notion of formal expressions.

The set $\mathcal{E}(A)$ of *formal expressions* over A is freely generated by the following grammar

$$e ::= x_i \mid c_a \mid e \cdot e$$

where $x_0, x_1, \ldots$ is an infinite amount of variables and c_a is a constant for every $a \in A$. We normally just write a in place of c_a and $x, y, z, \ldots$ as metavariables for $x_i, i \in \mathbb{N}$.

We write $e[e'/x]$ for the formal expression obtained by substituting all free occurrences of x in e by e'.

Proposition 2.2 (Combinatory completeness) *(i) Let A be a combinatory algebra. For any $e \in \mathcal{E}(A)$ exists a formal expression $\lambda^* x.e \in \mathcal{E}(A)$, whose variables are just those of e excluding x, such that*

$$(\lambda^* x.e)a = e[a/x]$$

for all $a \in A$.

(ii) If $(A, \cdot)$ is a set together with a binary operation satisfying the above property, it is a combinatory algebra.

Proof: (i) See [Lon94, Chapter 1.1].

(ii) Set $k = \lambda^* x.\lambda^* y.y$ and $s = \lambda^* x.\lambda^* y.\lambda^* z.(xz)(yz)$. □

By combinatory completeness it is easy to show that there are elements of A (called *combinators*) **i**, **pair**, **fst**, **snd**, **cond**, **true**, **false**, $\overline{\mathbf{n}}$ (for every $n \in \mathbb{N}$), **succ**, **pred** and **rec** which allow us to encode the notions of identity, pairing and projections, booleans and conditionals, natural numbers and primitive recursion in any combinatory algebra. More explicitly, they have the following properties:

$$\begin{array}{ll}
\mathbf{i}\, a = a & \\
\mathbf{fst}(\mathbf{pair}\, a\, b) = a & \mathbf{snd}(\mathbf{pair}\, a\, b) = b \\
\mathbf{cond\ true}\, a\, b = a & \mathbf{cond\ false}\, a\, b = b \\
\mathbf{succ}\, \overline{\mathbf{n}} = \overline{\mathbf{n+1}} & \mathbf{pred}\, \overline{\mathbf{n+1}} = \overline{\mathbf{n}} \\
\mathbf{rec}\, a\, f\, \overline{\mathbf{0}} = a & \mathbf{rec}\, a\, f\, \overline{\mathbf{n+1}} = f(\mathbf{rec}\, a\, f\, \overline{\mathbf{n}})
\end{array}$$

for all elements $a, b, f \in A$ and for all $n \in \mathbb{N}$.

So in spite of the simplicity of its definition, the notion of combinatory algebras is rich enough to encode all partial recursive functions.

We now introduce the notion of equivalence of combinatory algebras as described in [Lon94] for the special case of total combinatory algebras.

Definition 2.3 *Two combinatory algebras A and B are* equivalent*, written $A \simeq B$, if and only if there exist functions $\gamma \colon A \to B$ and $\delta \colon B \to A$ and elements $r_\delta, t, t' \in A$, $r_\gamma, s, s' \in B$, satisfying*

$$\gamma(a \cdot a') = r_\gamma \cdot \gamma(a) \cdot \gamma(a') \qquad \delta(b \cdot b') = r_\delta \cdot \delta(b) \cdot \delta(b')$$

$$t \cdot \gamma\big(\delta(a)\big) = a \qquad t' \cdot a = \gamma\big(\delta(a)\big)$$

$$s \cdot \delta\big(\gamma(b)\big) = b \qquad s' \cdot b = \delta\big(\gamma(b)\big)$$

for all elements $a, a' \in A$ and $b, b' \in B$.

2.2 Categories of Assemblies and Modest Sets

In order to provide models for (sequential) programming languages such as PCF or FPC (see Section 3), we will now define a *category of modest sets* over a (for this section fixed) combinatory algebra A. Instead of giving the definition immediately, we first introduce the *category of assemblies* and then find the modest sets as a full subcategory.

For a thorough investigation of the relation between the category of modest sets, the category of assemblies and the realisability topos see [Lon94].

The idea behind modest sets, going back to Kleene [Kle45], is to think of the combinatory algebra as a kind of untyped universe of computations (like some kind of machine language) and of the category of modest sets as a universe of data-types whose elements are represented by elements of the combinatory algebra.

Definition 2.4 *(i) An* assembly *X on A consists of an underlying set $|X|$ together with a nonempty subset $\|x\|_X \subseteq A$ for every $x \in |X|$. We say that an element $a \in A$* realises *some $x \in |X|$, and write $a \Vdash_X x$, if $a \in \|x\|_X$.*

(ii) A morphism between assemblies X and Y is a function $f\colon |X| \to |Y|$ which is tracked *by an element r of A, written $r \Vdash f$, i.e. if*

$$a \Vdash_X x \implies r \cdot a \Vdash_Y f(x)$$

for any $x \in |X|$ and $a \in A$.

We simply write $\|x\|$ and $a \Vdash x$ if the assembly is clear from the context.

Using combinatory completeness it is a simple matter to show that assemblies and morphism between them form a category, denoted $\mathcal{A}sm(A)$.

Definition 2.5 *An assembly X is called a* modest set*, if*

$$\|x\| \cap \|x'\| \neq \emptyset \implies x = x'$$

for all $x, x' \in X$.

Obviously, the category of modest sets, or simply the modest sets, denoted by $\mathcal{M}od(A)$, forms a full subcategory of the category of assemblies.

Furthermore, $\mathcal{M}od(A)$ has a terminal object, a natural numbers object, binary products and exponentials. There is even an object representing the combinatory algebra A itself. More explicitly, we define

$$\begin{array}{ll} |1| = \{\emptyset\} & \|\emptyset\|_1 = A \\ |N| = \mathbb{N} & \|n\|_N = \{\overline{\mathbf{n}}\} \\ |A| = A & \|a\|_A = \{a\} \end{array}$$

and for any two modest sets X and Y

$$\begin{array}{rr} |X \times Y| = |X| \times |Y| & \|(x,y)\|_{X\times Y} = \{\mathbf{pair}\, a\, b \mid a \in \|x\|_X, b \in \|y\|_Y\} \\ |Y^X| = \mathcal{M}od(X,Y) & \|f\|_{Y^X} = \{r \in A \mid r \text{ tracks } f\} \end{array}$$

So the category of modest sets, as well as the category of assemblies, is a cartesian closed category with natural numbers object. To avoid ambiguity we will sometimes write 1_A or N_A.

Some objects in the category of modest sets, namely the *projective objects*, play an important role in this diploma thesis.

Definition 2.6 *An object P of an arbitrary category is called* projective*, if for every regular epimorphism $e\colon Q \to R$ and every morphism $f\colon P \to R$ exists a (not necessarily unique) morphism $g\colon P \to Q$ such that $f = e \circ g$.*

$$\begin{tikzcd} P \arrow[r, "\forall f"] \arrow[d, "\exists g"'] & R \\ Q \arrow[ur, "\forall e"'] & \end{tikzcd}$$

In the case of the category of assemblies, projective objects can be characterised in a nice way.

Proposition 2.7 *An assembly is projective in $\mathcal{A}sm(A)$ if and only if it is isomorphic to an assembly in which every element is realised by precisely one element. An assembly of this form is called* partitioned.

Proof: see [Lon94, Chapter 2] □

Of course, any modest set which is projective is isomorphic to a partitioned modest set.

2.3 Dominances and Divergences

As we have seen, the category of modest sets has a rich categorical structure. However, in order to give rise to (realisability) models for PCF, we need more. As PCF allows partial functions and recursion, a notion of 'partial maps' is needed.

A clean categorical approach is that of Rosolini [Ros86], namely *dominances.* The idea behind the definition of dominances is, like it is traditionally done in domain theory, to view partial maps from X to Y as total maps from X into the 'lifted' object $Y_\perp$. The lifted object (if it is a set) consists of Y together with an additional element $\perp$, representing undefined computation. Moreover, not arbitrary partial maps are considered, but only those whose domains of definition are recursively enumerable sets.

Now a dominances is a piece of extra structure on a category, namely an object Σ, representing the lifted terminal object, together with a morphism $\top\colon 1 \to \Sigma$ (*classifying mono*), determining the class of 'partial morphisms'. Moreover, every dominance gives rise to a lifting monad whose functor is called the *lifting functor.*

However, in the case of modest sets, all dominances are already determined by a '*divergence*', a subset of the combinatory algebra indicating what elements are to be considered as diverging computation (cf. [Lon94, Chapter 4] for further details).

Definition 2.8 *A* divergence *on a combinatory algebra A is a nonempty proper right ideal $D \subseteq A$, i.e. $a \cdot b \in D$ for all $a \in D$ and $b \in A$.*

The corresponding dominance can then be defined in terms of divergence:

$$|\Sigma_D| = \{\perp, \top\}$$

$$\|\top\| = \{\mathbf{i}\} \qquad \|\perp\| = D$$

$$\begin{array}{rcl} \top_D\colon 1 & \to & \sigma \\ * & \mapsto & \top \end{array} \qquad \text{realised by } \lambda^* x.\,\mathbf{i}$$

where $\mathbf{i} := \lambda^* x.x$ (see the list of combinators on Page 10).

We define also the lifting functor but it can be shown that it is exactly the functor of the lifting monad coming from the above dominance.

Definition 2.9 *Given a divergence D on A a lifting functor*

$$L = (\cdot)_\perp \colon \mathcal{M}od(A) \to \mathcal{M}od(A)$$

can be defined as follows:

(i) for any modest set X, the object $L(X) = X_\perp$ is given by

$$\begin{aligned} |X_\perp| &= |X| \uplus \{\perp\} \\ \|x\|_{X_\perp} &= \{\mathbf{pair}\,\mathbf{i}\ a \mid a \in \|x\|_X\} \\ \|\perp\|_{X_\perp} &= \{\mathbf{pair}\,d\,a \mid d \in D, a \in A\} \end{aligned}$$

(ii) for any morphism $f \colon X \to Y$, tracked by some $t \in A$, the lifted morphism $L(f) = f_\perp$ is obtained by the lifting of the underlying function and is tracked by $\lambda^* x.\, \mathbf{pair}(\mathbf{fst}\,x)\big(t \cdot (\mathbf{snd}\,x)\big)$.

We can now extend the notion of equivalence of combinatory algebras to equivalence of combinatory algebras equipped with divergences.

Definition 2.10 *Combinatory algebras with divergences, (A, D) and (B, E), are* equivalent*, written $(A, D) \simeq (B, E)$, if the combinatory algebras are equivalent via functions $\gamma \colon A \to B$ and $\delta \colon B \to A$ and if there are elements $r, r' \in A$ and $s, s' \in B$ such that*

$$r \cdot \mathbf{i} = r' \cdot \mathbf{i} = \mathbf{i} \qquad r \cdot d \in \delta(E) \qquad r' \cdot d' \in D$$

$$s \cdot \mathbf{i} = s' \cdot \mathbf{i} = \mathbf{i} \qquad s \cdot e \in \gamma(D) \qquad s' \cdot e' \in E$$

for all $d \in D$, $d' \in \delta(E)$, $e \in E$ and $e' \in \gamma(D)$.

The main reason for this extension is that it reflects exactly the notion of equivalence between categories equipped with dominances:

Proposition 2.11 *For combinatory algebras with divergences, (A, D) and (B, E), the following statements are equivalent:*

(i) $(A, D) \simeq (B, E)$

(ii) $\mathcal{M}od(A) \simeq \mathcal{M}od(B)$ via a functor $F \colon A \to B$ such that $F(\top_D) \cong \top_E$.

Proof: See [Lon94, Chapter 4] □

2.4 PCF Models

We do not go into details here but will just mention that every category of modest sets, such that $N_\perp$ satisfies a certain well-completeness axiom[1], is a PCF model. This is described in detail in [LS97] and [Lon94].

For the two realisability models we are interested in, it is indeed possible to show that they satisfy this property. However, the notion of well-completeness is not needed to prove our results, so we do not go into details here.

In order to interpret PCF, usually only those objects (types) are used, which can be obtained from the natural numbers object by lifting and exponentiation. Among those, the *pure types* play a special role.

Definition 2.12 *The* pure types *are objects* $\underline{0}, \underline{1}, \underline{2}, \ldots$ *of* $\mathcal{A}sm(A)$ *inductively defined by*

$$\underline{0} := N \qquad \underline{n+1} := \underline{n} \to N_\perp := (N_\perp)^{\underline{n}}$$

[1]i.e. $N_\perp^i$ is an isomorphism, where $i\colon \omega \to \bar{\omega}$ is the inclusion of the initial into the terminal lifting algebra

3 The MRS Model

We sketch the definition of the combinatory algebra $\mathcal{U}$, going back to Marz, Rohr and Streicher, presented in [MRS99]. In a nutshell, $\mathcal{U}$ is a quotient of (the closed terms of) the language $\mathcal{L}$ which is a simple extension of untyped lambda calculus. The ideas we are going to present in this section are all taken from [MRS99] as well as from A. Rohr's doctoral dissertation (soon to appear) where all this will be treated in detail.

As it is pointed out in [MRS99], $\mathcal{U}$ can be found as a subalgebra of a combinatory algebra U whose underlying set is the solution of the domain equation $U \simeq 1 + (U \to U)$ in the category $\mathcal{SD}$ of sequential domains, the relational model due to M. Marz [Mar00].

$\mathcal{L}$ is 'the' language fitting to this domain equation, namely untyped lambda calculus with a single constant element (thought of as an error element) and a syntactic construction deciding whether a term is this element or whether it is a function.

$\mathcal{U}$ is then defined to be the quotient $\mathcal{L}/\!\sim$ of $\mathcal{L}$ by observable equivalence. It is easy to show by combinatory completeness that $\mathcal{U}$ is a combinatory algebra.

In A. Rohr's thesis will be shown that the category of modest sets over $\mathcal{U}$ is a universal and, therefore, fully abstract model for PCF. Furthermore, it follows from a result of that thesis that every object occurring as denotation of some PCF type is projective.

In Section 5 another programming language is used, namely FPC, which is closely related to the already mentioned category $\mathcal{SD}$. $\mathcal{SD}$ is a cartesian closed cpo-enriched category with a natural numbers object, sums, bilimits and lifting. FPC is an extension[2] of PCF with syntactic constructions reflecting these categorical properties. More explicitly, FPC has n-ary sums and recursive types. As M. Marz points out in his doctoral dissertation[3] [Mar00], $\mathcal{SD}$ is a fully abstract model not only for PCF but also for FPC. Moreover, every realisability model is a model for FPC. So it makes sense in any realisability model to speak of FPC definable morphism.

[2]It is strongly believed but not yet shown that this extension is conservative (for PCF types).

[3]In M. Marz's thesis, not FPC is treated, but a conservative extension of it, containing some more type constructors and call-by-value features, called SFL.

4 Game Semantics

We first define a combinatory algebra $\mathcal{A}$ and later identify a subalgebra $\mathcal{A}_{wb}$ of well-bracketed strategies, giving rise to a fully abstract model for PCF. We establish some properties of the categories $\mathcal{M}od(\mathcal{A})$ and $\mathcal{M}od(\mathcal{A}_{wb})$ and will find good representations of the objects N and $N \to N_\perp$.

Finally we restrict $\mathcal{A}$ and $\mathcal{A}_{wb}$ to its effective parts $\mathcal{A}_{\mathit{eff}}$ and $\mathcal{A}_{wb,\mathit{eff}}$, respectively, and obtain a universal model for PCF.

4.1 The Combinatory Algebra of History-free Strategies

The combinatory algebra $\mathcal{A}$ was first constructed by Abramsky and Longley and presented in [Lon98a] and is equivalent to van Oosten's algebra $\mathcal{B}$, see [vO97] and [AL99].

We define the underlying set of $\mathcal{A}$ to be the set of partial functions from M to M, where M is a set of *moves* defined as follows:

$M = C \times T$ with $T = \mathbb{N} \cup \{*\}$, the set of *tokens*, and C, the set of *contexts*, freely generated by the grammar

$$c ::= [.] \mid Lc \mid Rc \mid \vartheta(i, c)$$

for $i \in \mathbb{N}$.

For simplicity of notation, we write $R(i, m)$, $R_i(m)$ or even $R_i m$ instead of $R\vartheta(i, m)$.

An element m of M is written $c[t]$ or simply t, if c is the empty context. The set M can be viewed as being essentially the natural numbers endowed with injective coding functions $[L, R] \colon M + M \to M$ and $\vartheta \colon \mathbb{N} \times M \to M$. Moreover, we fix a bijective encoding $\kappa \colon M \to \mathbb{N}$.

Definition 4.1 *Given an element $f \in \mathcal{A}$, a finite sequence $s = m_1, \ldots, m_{2r}$ of elements in M is called a* dialogue with f *if and only if $m_{2i} = f(m_{2i-1})$ for $i = 1, \ldots, r$.*

We now define the application $\bullet \colon \mathcal{A} \times \mathcal{A} \to \mathcal{A}$ as follows:

Definition 4.2 *For f and g in $\mathcal{A}$,*

$$f \bullet g(m) = m'$$

holds, if and only if there exists a (unique) dialogue $b_2, \ldots, b_{2r-1}$ with g and natural numbers $i_2, \ldots, i_{2r-1}$ such that

$$Lm, R_{i_2} b_2, \ldots, R_{i_{2r-1}} b_{2r-1}, Lm$$

is a dialogue with f.

The idea behind the definition of this application is to consider f as a partial endofunction of $M + \mathbb{N} \times M$ (via the coding functions $[L, R]$ and ϑ) and to feed an input to the 'left side' of f. If the result is again on the left, return it as the output, otherwise apply g to it and continue from the beginning. The encoding via $\mathbb{N}$ is needed in order to give f access to infinitely many copies of g.

It is also convenient to visualise f as a box with one input wire and one output wire, each capable of carrying one element of M.

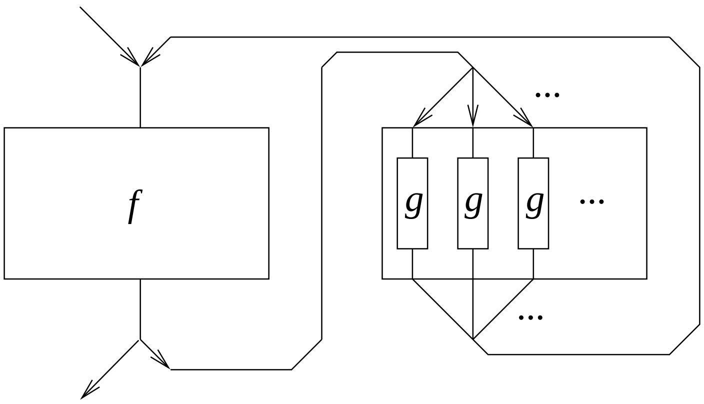

Figure 1: The application $f \bullet g$

If we want to compute some $\alpha \bullet \beta(m)$ in the following, we will often write the 'computing dialogue' in the form:

$$Lm \xrightarrow{\alpha} R_i m \xrightarrow{\beta} R_i \beta(m') \xrightarrow{\alpha} \cdots \xrightarrow{\alpha} L\big(\alpha \bullet \beta(m)\big)$$

Proposition 4.3 *The algebra $\mathcal{A}$ is a combinatory algebra.*

Proof: As we do not need the elements k and s of $\mathcal{A}$ explicitly (in fact, s is very difficult to obtain), we just refer to [AL99, Section 1]. $\square$

Example 4.4 *The combinator* $\mathbf{i}$ *of* $\mathcal{A}$ *can be explicitly given by*

$$\mathbf{i}(Lm) = R_0 m \qquad \mathbf{i}(R_0 m) = Lm$$

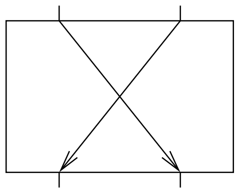

Figure 2: the combinator $\mathbf{i}$

We will show now what the natural numbers object of $\mathcal{M}od(\mathcal{A})$ looks like. We define $N = N_{\mathcal{A}}$ by

$$|N| = \mathbb{N} \qquad \|n\| = \{\varrho_n\}$$

where ϱ_n maps $*$ to n and is undefined elsewhere.

Proposition 4.5 *The object $N_{\mathcal{A}}$ is the natural number object of $\mathcal{M}od(\mathcal{A})$.*

Proof: Zero is realised by $\zeta\colon L* \mapsto L0$ and the successor function by

$$\begin{array}{rcl} \sigma\colon L* & \mapsto & R_0 * \\ R_0 n & \mapsto & L(n+1) \end{array}$$

If we have $1 \xrightarrow{g} X \xrightarrow{f} X$ with $\gamma \Vdash g$ and $\varphi \Vdash f$, the (unique) morphism $u\colon N \to X$ is defined by $n \mapsto (f^n \circ g)(*)$ and is realised by

$$\begin{array}{rcll} \epsilon\colon Lm & \mapsto & R_{\kappa(m)} * & \text{for all } m \in M \\ R_i n & \mapsto & L\Big(\underbrace{\varphi \bullet (\varphi \bullet \cdots \bullet (\varphi}_{n \text{ times}} \bullet (\gamma \bullet \mathbf{i})) \cdots)\Big)\big(\kappa^{-1}(i)\big) & \text{for all } i \in \mathbb{N} \end{array}$$

$\square$

We fix a divergence on $\mathcal{A}$ and give an explicit representation of the object $N_\perp$. We set $D = D_{\mathcal{A}} = \{\perp_{\mathcal{A}}\}$, where $\perp_{\mathcal{A}}$ is the everywhere undefined function on M, and define $N_\perp$ by

$$|N_\perp| = |N| \cup \{\perp\}, \qquad \|n\|_{N_\perp} = \{\varrho_n\}, \qquad \|\perp\|_{N_\perp} = \{\varrho_\perp\}$$

where $\varrho_\perp = \perp_{\mathcal{A}}$.

It is straightforward but tedious to check that $N_\perp$ is indeed the object obtained by lifting N via the from D induced lifting functor.

We are now ready to find a good representation of the object $\underline{1}$. The next lemma and proposition will show that $\underline{1}$ is in fact isomorphic to $\mathcal{A}$ itself, regarded as a modest set. As the proofs are typical for the techniques used in this paper they will be very detailed. Further proofs of similar results will be much shorter and details left to the reader.

Lemma 4.6 *Let β realise some element f of $\underline{1}$. Then for every natural number n there are elements $i_1, \ldots, i_k$ $(k = 0, 1, \ldots)$ of pairwise distinct natural numbers such that*

(i) if $f(n)$ is defined, there is a dialogue with β of the form[4]

$$L* \xrightarrow{\beta} R_{i_1}* \xrightarrow{\varrho_n} R_{i_1}n \xrightarrow{\beta} R_{i_2}* \xrightarrow{\varrho_n} \cdots \xrightarrow{\varrho_n} R_{i_k}n \xrightarrow{\beta} Lf(n)$$

(ii) if $f(n)$ diverges, there is a (similar) dialogue with β which either runs forever or is of the form

$$L* \xrightarrow{\beta} R_{i_1}* \xrightarrow{\varrho_n} R_{i_1}n \xrightarrow{\beta} R_{i_2}* \xrightarrow{\varrho_n} \cdots \xrightarrow{\beta} R_{i_k}* \xrightarrow{\varrho_n} R_{i_k}n$$

and $\beta(R_{i_k}n)$ either diverges or is not of the form R_i* for any $i \in \mathbb{N}$.

Proof: By definition of the realisability relation

$$\begin{aligned} \beta \Vdash_{\underline{1}} f &\iff \forall n \in \mathbb{N}. \beta \bullet \varrho_n \Vdash_{\underline{1}} f(n) \\ &\iff \forall n \in \mathbb{N}. \beta \bullet \varrho_n = \varrho_{f(n)} \\ &\implies \forall n \in \mathbb{N}. \beta \bullet \varrho_n(*) \simeq f(n) \end{aligned}$$

Computing $\beta \bullet \varrho_n(*)$, we get exactly the claimed sequences. It rests to argue for what reasons a computation may diverge.

There are only two reasons: First, the computation may run forever. Second, β may diverge at one point or return something which causes ϱ_n or the algorithm to diverge. But this is always the case if β returns anything different from R_i* (note that returning Ln' for some $n' \in \mathbb{N}$ is not possible, too, because this means that $f(n)$ is defined). $\square$

[4]note that this includes the case $\beta(L*) = Lf(n)$ where f is constant

Proposition 4.7 *The objects $\underline{1}$ and $\mathcal{A}$ of $\mathcal{M}od(\mathcal{A})$ are isomorphic.*

Proof: We define

$$\begin{array}{rcl} F\colon \mathcal{A} & \to & \underline{1} \\ f & \mapsto & \kappa \circ f \circ \kappa^{-1} \end{array} \qquad \text{and} \qquad \begin{array}{rcl} G\colon \underline{1} & \to & \mathcal{A} \\ g & \mapsto & \kappa^{-1} \circ g \circ \kappa \end{array}$$

using the bijective encoding $\kappa\colon M \to \mathbb{N}$.

Obviously, the two mappings are inverse to each other and it rests to show that they are realisable.

If $\alpha \in \mathcal{A}$ is a realiser of F, then for every $f \Vdash_{\mathcal{A}} f$ we have $\alpha \bullet f \Vdash_{\underline{1}} \kappa \circ f \circ \kappa^{-1}$. But

$$\begin{array}{rcl} \alpha \bullet f \Vdash_{\underline{1}} \kappa \circ f \circ \kappa^{-1} & \iff & \forall n \in \mathbb{N}. \alpha \bullet f \bullet \varrho_n \Vdash_{\underline{1}} \kappa \circ f \circ \kappa^{-1}(n) \\ & \iff & \forall n \in \mathbb{N}. \alpha \bullet f \bullet \varrho_n = \varrho_{\kappa \circ f \circ \kappa^{-1}(n)} \\ & \iff & \forall n \in \mathbb{N}. \alpha \bullet f \bullet \varrho_n(*) \simeq \kappa \circ f \circ \kappa^{-1}(n) \quad \wedge \\ & & \forall n \in \mathbb{N}. \forall m \in M \setminus \{*\}. \alpha \bullet f \bullet \varrho_n(m)\uparrow \end{array}$$

We define such an α by:

$$\begin{array}{rrcll} \alpha\colon & LL* & \mapsto & LR_0* & \\ & LR_0 n & \mapsto & R_0(\kappa^{-1}(n)) & \text{for all } n \in \mathbb{N} \\ & R_0 m & \mapsto & LL\kappa(m) & \text{for all } m \in M \end{array}$$

and check that it fullfills the last formula of the above equivalence.

Be $f \in A$ and $n \in \mathbb{N}$. If $f\big(\kappa^{-1}(n)\big)$ is defined we obtain[5]

$$L* \xrightarrow{\alpha \bullet f} R_0 * \xrightarrow{\varrho_n} R_0 n \xrightarrow{\alpha \bullet f} L\big(\kappa \circ f \circ \kappa^{-1}(n)\big)$$

and if $f\big(\kappa^{-1}(n)\big)$ diverges the latter sequence diverges at the third step. So for all natural numbers

$$\alpha \bullet f \bullet \varrho_n(*) \simeq \kappa \circ f \circ \kappa^{-1}(n)$$

On the other hand, $\alpha \bullet f \bullet \varrho_n(m)$ is clearly not defined if $m \neq *$ because α then diverges at LLm.

[5]The reader should check the last arrow himself.

Conversely, if α' is a realiser for G, then $\alpha' \bullet \beta \Vdash_{\mathcal{A}} \kappa^{-1} \circ g \circ \kappa$ for every $\beta \Vdash_{\underline{1}} g$. But

$$\begin{aligned} \alpha' \bullet \beta \Vdash_{\mathcal{A}} \kappa^{-1} \circ g \circ \kappa &\iff \alpha' \bullet \beta = \kappa^{-1} \circ g \circ \kappa \\ &\iff \forall m \in M. \alpha' \bullet \beta(m) \simeq \kappa^{-1} \circ g \circ \kappa(m) \end{aligned}$$

We define such an α' by:

$$\begin{array}{rcll} \alpha' \colon Lm & \mapsto & R_{\kappa(m)} L* & \text{for every } m \in M \\ R_n R_i * & \mapsto & R_n R_i n & \text{for every } i \in \mathbb{N} \\ R_n L n' & \mapsto & L\kappa^{-1}(n') & \text{for all } n, n' \in \mathbb{N} \end{array}$$

Let β be a realiser for some $g \colon N \to N_\perp$ and m some element of M. We check that $\alpha' \bullet \beta(m) \simeq \kappa^{-1} \circ g \circ \kappa(m)$.

First, if $g\big(\kappa(m)\big)$ is defined, we obtain the following dialogue by applying the last lemma:

$$Lm \xrightarrow{\alpha'} R_{\kappa(m)} L* \xrightarrow{\beta} R_{\kappa(m)} R_{i_0} * \xrightarrow{\alpha'} R_{\kappa(m)} R_{i_0} \kappa(m) \xrightarrow{\beta} R_{\kappa(m)} R_{i_1} * \xrightarrow{\alpha'} \cdots$$

$$\cdots \xrightarrow{\alpha'} R_{\kappa(m)} R_{i_k} \kappa(m) \xrightarrow{\beta} R_{\kappa(m)} Lg\big(\kappa(m)\big) \xrightarrow{\alpha'} L\big(\kappa^{-1} \circ g \circ \kappa(m)\big)$$

So $\alpha' \bullet \beta(m) = \kappa^{-1} \circ g \circ \kappa(m)$ holds.

Second, if $g\big(\kappa(m)\big)$ is undefined, $\alpha' \bullet \beta(m)$ diverges as well, because either the computation runs forever or at one point β returns something not of the form R_i*, which causes α' or the algorithm to diverge. □

Corollary 4.8 *It follows immediately that $\underline{2} \simeq N_\perp^{\mathcal{A}}$ in $\mathcal{M}od(\mathcal{A})$.*

Subsequently, if not otherwise stated, $\underline{1}_{\mathcal{A}}$ and $\underline{2}_{\mathcal{A}}$ stand for $\mathcal{A}$ and $N_\perp^{\mathcal{A}}$, respectively. We will investigate now, what realisers of $\underline{2}$ look like. There is an observation similar to Lemma 4.6:

Lemma 4.9 *Let α be a realiser for some element F of $\underline{2}$. Then for every element f of $\underline{1}$ exist pairwise distinct natural numbers $i_1, \ldots, i_k$ and elements $m_1, \ldots, m_k$ ($k = 0, 1, \ldots$) of M such that*

(i) if $F(f)$ is defined, there is a dialogue with α of the form

$$L* \xrightarrow{\alpha} R_{i_1} m_1 \xrightarrow{f} R_{i_1} f(m_1) \xrightarrow{\alpha} R_{i_2} m_2 \xrightarrow{f} \cdots \xrightarrow{f} R_{i_k} f(m_k) \xrightarrow{\alpha} LF(f)$$

(ii) if $F(f)$ diverges, there is a dialogue with α which is infinite or is of the form

$$L{*}\xrightarrow{\alpha}R_{i_1}m_1\xrightarrow{f}R_{i_1}f(m_1)\xrightarrow{\alpha}R_{i_2}m_2\xrightarrow{f}\cdots\xrightarrow{\alpha}R_{i_k}m_k\xrightarrow{f}R_{i_k}f(m_k)$$

and $\alpha\big(R_{i_k}f(m_k)\big)$ either diverges or is not of the form R_im' for any $i \in \mathbb{N}$ and $m' \in M$.

Proof: Like in Lemma 4.6, it suffices to unfold the definition of $\alpha \bullet \beta \Vdash_N F(f)$ and to convince yourself that in the case of divergence the claimed possibilities are the only ones. $\square$

4.2 A Subalgebra of Well-bracketed Strategies

We will now identify a subalgebra $\mathcal{A}_{wb}$ of $\mathcal{A}$, due to Abramsky and Longley, giving rise to a realisability model equivalent to, as shown in [AL99], the history-free game model of Abramsky, Jagadeesan and Malacaria [AJM94], hence to a fully abstract model for PCF.

To define the subalgebra $\mathcal{A}_{wb}$ we have to introduce the notion of well-bracketedness.

Definition 4.10 *A move $c[t]$ of M is called a* question*, if $t = *$, an* answer *otherwise. We say that a move $c[t]$* matches *a move $c'[t']$ if $c[t]$ is an answer, $c'[t']$ is a question and $c = c'$.*

Definition 4.11 *(i) The* balanced sequences *of moves in M are defined inductively as follows:*

- *the empty sequence ϵ is balanced*
- *if σ is balanced and a matches q then $q\,\sigma\,a$ is balanced.*
- *if σ and τ are balanced sequences then so is their concatenation $\sigma\,\tau$.*

(ii) An (infinite) sequence is well-bracketed *if every finite prefix is prefix of some balanced sequence.*

(iii) A sequence $m_1, m_2, \ldots$ contains a bracketing violation *at position r (or at the move m_r), if $m_1, \ldots, m_{r-1}$ is well-bracketed but $m_1, \ldots, m_r$ is not.*

(iv) A function $f \in \mathcal{A}$ is well-bracketed*, if no dialogue with f contains a bracketing violation at an even position.*

Remark 4.12 *(i) There is at most one bracketing violation for every sequence.*

(ii) A sequence $m_1, \ldots, m_k$ has a bracketing violation at position r if and only if either $r = 1$ and m_1 is an answer or there is an $r' < r$ such that $m_{r'}$ is a question, m_r is an answer not matching $m_{r'}$, and $m_{r'+1}, \ldots, m_{r-1}$ is balanced.

We define $\mathcal{A}_{wb}$ to be the subset of of well-bracketed elements of $\mathcal{A}$.

Proposition 4.13 *The set $\mathcal{A}_{wb} := \{f \in \mathcal{A} \mid f \text{ is well-bracketed}\}$ together with the restriction of $\bullet\colon \mathcal{A} \to \mathcal{A}$ to $\mathcal{A}_{wb}$ is a combinatory algebra.*

Proof: see [AL99, Section 3] □

There is a nice characterisation of well-bracketed elements of $\mathcal{A}$ which turns out to be be very useful later on.

Definition 4.14 *A dialogue with f, $m_1, \ldots, m_{2k}$, is called* quick *if m_1 is a question and m_{2i+1} matches m_{2i}, for $0 \leq i \leq k-1$, whenever the latter is a question.*

Proposition 4.15 *$f \in \mathcal{A}$ is well-bracketed if and only if every quick dialogue with f is well-bracketed.*

Proof: Suppose f is well-bracketed and let $\sigma = m_1, \ldots, m_r$ be a quick dialogue with f. By assumption, σ contains no bracketing violation at an even position. So be there one at an odd position, say $k > 1$. By Remark 4.12, and because balanced-sequences are always of even length, there is a last unanswered question at an even position, $k' < k$, which is not matched by m_k. But as σ is quick, either $k'+1 = k$ and m_k matches $m_{k'}$, which is a contradiction to the assumption, or $k'+1 < k$ and $m_{k'+1}$ matches $m_{k'}$, so $m_{k'}$ cannot be last unanswered question.

Suppose now f is not well-bracketed. Then there exists a dialogue with f, $m_1, \ldots, m_n$ containing a bracketing violation at an even position, say k. That implies that there is a question $m_{k'}$, $k' < k$ such that m_k does not match $m_{k'}$ and $m_{k'+1}, \ldots, m_{k-1}$ is a balanced sequence. From this latter dialogue we now delete every balanced subsequence which starts at an odd position and obtain a sequence $m_{k'}, m'_1, \ldots, m'_{2l}, m_k$ which is of the form

$$c[*], c_1[*], c_1[n_1], \ldots, c_l[*], c_l[n_l], c'[n]$$

This is a quick dialogue with f which is not well-bracketed, as $c[*] = m_{k'}$ is not matched by $c'[n] = m_k$. □

We will now identify the natural numbers object in $\mathcal{M}od(\mathcal{A}_{wb})$ and investigate the objects $\underline{1}_{\mathcal{A}_{wb}}$ and $\underline{2}_{\mathcal{A}_{wb}}$.

As each unique realiser ϱ_n is trivially well-bracketed, $N_{\mathcal{A}}$ is a natural candidate for $N_{\mathcal{A}_{wb}}$. The proof that this is indeed the natural numbers object of $\mathcal{M}od(\mathcal{A}_{wb})$ is exactly the same as for Proposition 4.5 because all occurring realisers are well-bracketed.

We will now investigate $\underline{1}_{\mathcal{A}_{wb}}$. Obviously Lemma 4.6 still holds for $\mathcal{A}_{wb}$. But in contrast to $\mathcal{M}od(\mathcal{A})$ the object $\underline{1}$ seems not to be isomorphic to $\mathcal{A}_{wb}$ in $\mathcal{M}od(\mathcal{A}_{wb})$. Nevertheless, it is possible to (realisably) choose canonical realisers for elements of $\underline{1}_{\mathcal{A}_{wb}}$.

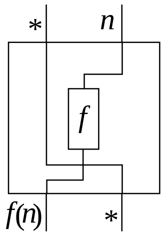

Figure 3: a canonical $\underline{1}$-realiser

It is easy to see that for any function $f : N \to N_\perp$ exists is a realiser $\beta \in \mathcal{A}_{wb}$ defined by

$$\begin{array}{rcll} \beta\colon L* & \mapsto & R_0* & \\ R_0 n & \mapsto & Lf(n) & \text{for every } n \in \mathbb{N} \end{array}$$

We call a realiser of this form *canonical* and show now that it is possible to realisably select those canonical realisers within $\mathcal{A}_{wb}$.

Proposition 4.16 *There is an element* $\mathbf{can}_{\underline{1}} \in \mathcal{A}_{wb}$ *such that for every* β*, realising some* $f \in \underline{1}$*,* $\mathbf{can}_{\underline{1}} \bullet \beta$ *is a canonical realiser for* f*.*

Proof: Let $\mathbf{can}_{\underline{1}}$ be defined by:

$$\begin{array}{rcll} \mathbf{can}_{\underline{1}}\colon LL* & \mapsto & LR_0* & \\ LR_0 n & \mapsto & R_n L* & \text{for all } n \in \mathbb{N} \\ R_n R_i * & \mapsto & R_n R_i n & \text{for all } i, n \in \mathbb{N} \\ R_n L n' & \mapsto & LLn' & \text{for all } n, n' \in \mathbb{N} \end{array}$$

(i) $\mathbf{can}_{\underline{1}}$ is well-bracketed: the only possible quick dialogues with $\mathbf{can}_{\underline{1}}$ are of the form:

$$LL* \xrightarrow{\mathbf{can}_{\underline{1}}} LR_0* \to LR_0n \xrightarrow{\mathbf{can}_{\underline{1}}} R_nL* \to R_nLn' \xrightarrow{\mathbf{can}_{\underline{1}}} LLn'$$

and

$$R_nR_i* \xrightarrow{\mathbf{can}_{\underline{1}}} R_nR_in$$

They are well-bracketed for any $n, n' \in \mathbb{N}$ and so is $\mathbf{can}_{\underline{1}}$ by Proposition 4.15.

(ii) $\mathbf{can}_{\underline{1}} \bullet \beta(L*) = R_0*$: by construction

(iii) $\mathbf{can}_{\underline{1}} \bullet \beta(R_0n) \simeq Lf(n)$. From Lemma 4.6 we know what β looks like. It is now a straightforward but tedious task to write down the computing dialogues, distinguishing the two cases $f(n)\downarrow$ and $f(n)\uparrow$.

(iv) $\mathbf{can}_{\underline{1}} \bullet \beta$ is not defined elsewhere, as $\mathbf{can}_{\underline{1}}$ is not. □

Corollary 4.17 *In $\mathcal{M}od(\mathcal{A}_{wb})$ the object $\underline{1}$ is isomorphic to the object $\tilde{1}$ defined by:*

$$|\tilde{1}| := |\underline{1}|$$

$$\|f\|_{\tilde{1}} = \{\beta \in \|f\|_{\underline{1}} \mid \beta \text{ is canonical}\}$$

Proof: The isomorphism is the identity realised by $\mathbf{i}$ and $\mathbf{can}_{\underline{1}}$, respectively. □

Remark 4.18 *Note that the canonical $\underline{1}$-realisers are unique, and so $\underline{1}$ is a partitioned modest set.*

From now on we will always identify $\underline{1}_{\mathcal{A}_{wb}}$ with $\tilde{1}$.

In view of Lemma 4.6 and the last proposition it is obvious that we can use the same element $\mathbf{can}_{\underline{1}}$ to translate realisers of elements of $\underline{1}_{\mathcal{A}}$ to well-bracketed ones. More explicitly we state the following proposition whose proof is identical to that of Proposition 4.16.

Proposition 4.19 *For all $\beta \in \mathcal{A}$, realising $f \in \underline{1}_{\mathcal{A}}$, $\mathbf{can}_{\underline{1}} \bullet \beta$ is a canonical (hence well-bracketed) realiser for f in $\mathcal{M}od(\mathcal{A})$.*

4.3 A Universal Model for PCF

However, as it is shown in [AL99], $\mathcal{A}_{wb,\mathit{eff}}$ yields a fully abstract but *not* universal model for PCF. To obtain a universal model, Abramsky and Longley restrict $\mathcal{A}_{wb}$ to its effective part.

More explicitly, they define $\mathcal{A}_{\mathit{eff}}$ to be the restriction of $\mathcal{A}$ to the maps from M to M corresponding to partial recursive functions from $\mathbb{N}$ to $\mathbb{N}$ via the encoding $\kappa\colon M \to \mathbb{N}$. Consequently they define $\mathcal{A}_{wb,\mathit{eff}}$ to be the subset of $\mathcal{A}_{\mathit{eff}}$ of effective well-bracketed elements and show that $\mathcal{A}_{wb,\mathit{eff}}$ yields a universal model for PCF.

It is obvious that all results of the last two sections still hold in the effective setting. For instance, all morphism, whose existens where proved in these sections, where explicitly given by sequential algorithms.

For the rest of this diploma thesis we will exclusively work with $\mathcal{A}_{\mathit{eff}}$ and $\mathcal{A}_{wb,\mathit{eff}}$ and will subsequently omit the subscript '*eff*'.

5 Type $\underline{2}$ is not projective in $\mathcal{M}od(\mathcal{A}_{wb,eff})$

We are now going to prove the main result of this diploma thesis, namely that the object $\underline{2}$ is not projective in $\mathcal{M}od(\mathcal{A}_{wb})$.

5.1 Outline of the Proof

If we assume that $\underline{2}$ *is* projective in $\mathcal{M}od(\mathcal{A}_{wb})$, then there is a realisable mapping from $\mathcal{A}_{wb}$ to $\mathcal{A}_{wb}$ 'shaving' $\underline{2}$-realisers to unique canonical ones. It is then possible to lift this 'shaver' to $\mathcal{A}$ by translating realisers of $\underline{2}_{\mathcal{A}}$ to realisers of $\underline{2}_{\mathcal{A}_{wb,eff}}$, shave them, and back-translate them to realisers of $\underline{2}_{\mathcal{A}}$.

$$\begin{tikzcd} \mathcal{A} \arrow[r, "\mathbf{sh}_{\mathcal{A}}"] \arrow[d, "\mathbf{tr}"'] & \mathcal{A} \\ \mathcal{A}_{wb} \arrow[r, "\mathbf{sh}_{\mathcal{A}_{wb}}"'] & \mathcal{A}_{wb} \arrow[u, "\mathbf{btr}"'] \end{tikzcd}$$

However, $\mathbf{sh}_{\mathcal{A}}$ gives rise to a right-inverse to the currying $p\colon N_{\perp}^{\mathcal{A}} \to \mathcal{A}$ of the application function $b\colon (f,g) \mapsto f \bullet g(*)$, which means that type $\underline{2} \simeq N_{\perp}^{\mathcal{A}}$ is a retract of $\underline{1} \simeq \mathcal{A}$ in $\mathcal{M}od(\mathcal{A})$. This can be shown to be impossible.

5.2 The Proof

First, we show that if $\underline{2}_{\mathcal{A}_{wb}}$ is projective, then a function $\mathbf{sh}_{\mathcal{A}_{wb}}$ on $\mathcal{A}_{wb}$ exists which chooses canonical realisers for elements of $\underline{2}_{\mathcal{A}_{wb}}$.

Lemma 5.1 *If $\underline{2}_{\mathcal{A}_{wb}}$ is projective then there exists a in $\mathcal{M}od(\mathcal{A}_{wb})$ realisable function $\mathbf{sh}_{\mathcal{A}_{wb}}\colon \mathcal{A}_{wb} \to \mathcal{A}_{wb}$ such that*

$$\alpha, \alpha' \Vdash F \Longrightarrow \mathbf{sh}_{\mathcal{A}_{wb}}(\alpha) = \mathbf{sh}_{\mathcal{A}_{wb}}(\alpha') \Vdash F$$

for all $F \in \underline{2}_{\mathcal{A}_{wb}}$.

Proof: We have to show that there is an element $\sigma \in \mathcal{A}_{wb}$ such that $\sigma \bullet \alpha = \sigma \bullet \alpha' \Vdash F$ whenever $\alpha, \alpha' \Vdash F$ for any $F \in \underline{2}_{\mathcal{A}_{wb}}$. If $\underline{2}_{\mathcal{A}_{wb}}$ is projective then an isomorphism $\varphi\colon \underline{2}_{\mathcal{A}_{wb}} \to X$ exists, where X is a partitioned modest set. Let β be a realiser of φ and β' a realiser of φ^{-1}. It is than straightforward to check that $\sigma_{\mathcal{A}_{wb}} := \lambda x.\beta' \bullet (\beta \bullet x)$ has the required property. □

We now construct the functions $\mathbf{tr}$ and $\mathbf{btr}$ in order to 'lift' $\mathbf{sh}_{\mathcal{A}_{wb}}$ to $\mathbf{sh}_{\mathcal{A}}$.

To define $\mathbf{tr}$, we first translate $\underline{2}_{\mathcal{A}}$-realisers to realisers of $N_{\perp}^{\mathcal{A}}$ (via the isomorphism of $\underline{2}_{\mathcal{A}}$ and $N_{\perp}^{\mathcal{A}}$) and then to well-bracketed ones, exploiting the unique representations of realisers of $\underline{1}_{\mathcal{A}_{wb}}$.

Lemma 5.2 *There exists a mapping* $\mathbf{tr}\colon \mathcal{A} \to \mathcal{A}_{wb}$ *with*

$$\alpha \Vdash_{\underline{2}_{\mathcal{A}}} F \implies \mathbf{tr}(\alpha) \Vdash_{\underline{2}_{\mathcal{A}_{wb}}} F$$

Proof: Let α be a realiser of some element F of $\underline{2}_{\mathcal{A}}$. In view of Proposition 4.7 it is clear that we can (realisably) transform α to an α', realising the following (to F corresponding) element of $N_{\perp}^{\mathcal{A}}$:

$$\begin{array}{rcl} F' \colon (M \rightharpoonup M) & \to & N_{\perp} \\ g & \mapsto & F(\kappa \circ g \circ \kappa^{-1}) \end{array}$$

Now we translate α' to a well-bracketed realiser α'' of F. We define α'' by:

$$\begin{array}{lcl} \alpha''(L*) & = & \begin{cases} Ln & \text{if } \alpha'(L*) = Ln \\ R(\langle i, \kappa(m)\rangle, L*) & \text{if } \alpha'(L*) = R_i m \end{cases} \\ \alpha''\Big(R\big(\langle \tau, i, n\rangle, R_0 *\big)\Big) & = & R\big(\langle \tau, i, n\rangle, R_0 n\big) \\ \alpha''\Big(R\big(\langle \tau, i, n\rangle, Ln'\big)\Big) & = & \begin{cases} R\big(\langle \tau, i, n, j, \kappa(m)\rangle, L*\big) & \text{if } \alpha'(R_i \kappa^{-1}(n')) = R_j m \\ Ln'' & \text{if } \alpha'(R_i \kappa^{-1}(n')) = Ln'' \end{cases} \end{array}$$

for $m \in M$, $n, n', n'', i, j \in \mathbb{N}$, $\tau \in \mathrm{Seq}(\mathbb{N})$ and an injective coding function $\langle \cdot \rangle\colon \mathrm{Seq}(\mathbb{N}) \to \mathbb{N}$.

Let f be an element of $\underline{1}_{\mathcal{A}_{wb}}$, realised by $\beta \in \mathcal{A}_{wb}$, and be $g = \kappa^{-1} \circ f \circ \kappa$ its corresponding element in $\mathcal{A}$ (where f is thought of as an element of $\underline{1}_{\mathcal{A}}$). We show that $\alpha'' \bullet \beta \Vdash F(f) = F'(g)$.

Let $F'(g)$ be defined. Then $\alpha' \bullet g$ must be defined as well and with help of Lemma 4.9 it is straightforward to check that $\alpha'' \bullet \beta(*) = F'(g)$.

On the other hand, if $F'(g)$ diverges, $\alpha'' \bullet \beta(*)$ diverges as well, as $\alpha'\big(R_{i_k} g(m_k)\big)$ is not equal to $R_i m$ for all $i \in \mathbb{N}$ and $m \in M$.

Furthermore it is clear that $\alpha'' \bullet \beta$ is not defined apart from $*$, as $\alpha''(Lm)$ is not defined for $m \neq *$, and that α'' is well-bracketed, as all quick dialogues with it are so.

So finally we can set $\mathbf{tr}(\alpha) := \alpha''$. $\square$

In order to define the back-translation we simply apply the fact that every $\underline{1}_{\mathcal{A}}$-realiser can be translated to a well-bracketed one (see Proposition 4.19).

Lemma 5.3 *There exists a mapping* $\mathbf{btr}\colon \mathcal{A}_{wb} \to \mathcal{A}$ *with the following property:*

$$\alpha \Vdash_{\underline{2}_{\mathcal{A}_{wb}}} F \implies \mathbf{btr}(\alpha) \Vdash_{\underline{2}_{\mathcal{A}}} F$$

Proof: We simply define $\alpha' := \mathbf{btr}(\alpha) := \lambda^* b.\alpha \bullet (\mathbf{can}_{\underline{1}} \bullet b)$.

Let α realise some $F \in \underline{2}_{\mathcal{A}_{wb}}$. It follows immediately that α' realises F in $\mathcal{M}od(\mathcal{A})$: if β is a realiser of some $f \in \underline{1}_{\mathcal{A}}$ then $\alpha' \bullet \beta = \alpha \bullet (\mathbf{can}_{\underline{1}} \bullet \beta) \simeq \varrho_{F(f)}$ as $\mathbf{can}_{\underline{1}} \bullet \beta$ is well-bracketed and realises f. Consequently, $\mathbf{btr}(\alpha)$ realises F in $\mathcal{M}od(\mathcal{A})$. □

We now define $\mathbf{sh}_{\mathcal{A}}$ as

$$\mathbf{sh}_{\mathcal{A}} := \mathbf{btr} \circ \mathbf{sh}_{\mathcal{A}_{wb}} \circ \mathbf{tr}$$

and obtain a map from $\mathcal{A}$ to $\mathcal{A}$, which picks out a single realiser for every element of $\underline{2}_{\mathcal{A}}$.

It is now easy to see that $\mathbf{sh}_{\mathcal{A}}$ is FPC definable and hence a morphism in $\mathcal{M}od(\mathcal{A})$: one can construct the combinatory algebra $\mathcal{A}$ as a first-order datatype of FPC and program the application $\bullet$ (being a sequential algorithm), in FPC, as well as the two maps $\mathbf{tr}$ and $\mathbf{btr}$ (viewed as programs of type $\mathcal{A} \to \mathcal{A}$). Furthermore, the shaver $\mathbf{sh}_{\mathcal{A}_{wb}}$ is FPC definable: its realiser $\sigma_{\mathcal{A}_{wb}} \in \mathcal{A}_{wb}$ tracks a functional of type $\mathcal{A} \to \mathcal{A}$ in $\mathcal{M}od(\mathcal{A})$ whose restriction to $\mathcal{A}_{wb}$ is again $\mathbf{sh}_{\mathcal{A}_{wb}}$. Consequently, as this functional is realisable in the universal model for FPC, $\mathcal{M}od(\mathcal{A}_{wb})$, it is FPC definable. Hence the composition of $\mathbf{tr}$, $\mathbf{sh}_{\mathcal{A}_{wb}}$ and $\mathbf{btr}$ is FPC definable.

We define the map p by

$$\begin{array}{rcl} p\colon \mathcal{A} & \to & N_{\perp}^{\mathcal{A}} \\ f & \mapsto & \big(g \mapsto f \bullet g(*)\big) \end{array}$$

and show that the (realiser of) $\mathbf{sh}_{\mathcal{A}}$ gives rise to a right-inverse of p. We will later show that this leads to a contradiction.

Lemma 5.4 *If there exists an element* $\sigma \in \mathcal{A}$ *with the property that*

$$\alpha, \alpha' \Vdash_{\underline{2}_{\mathcal{A}}} \implies \sigma \bullet \alpha = \sigma \bullet \alpha' \Vdash_{\underline{2}_{\mathcal{A}}} F$$

then p *has a right-inverse.*

Proof: We define the right-inverse $e\colon N_\perp^{\mathcal{A}} \to \mathcal{A}$ by setting $e(F) = \sigma \bullet \alpha$ for any α realising F. Clearly e is well-defined and realised by σ. All we have to show is that $p \bullet e(F)(g) = F(g)$ holds for all F of $N_\perp^{\mathcal{A}}$ and all g of $\mathcal{A}$. But

$$\begin{aligned} p \bullet e(F)(g) &= p\big(e(F)\big)(g) \\ &= e(F) \bullet g(*) \\ &= \mathbf{sh}(\alpha) \bullet g(*) \end{aligned}$$

for some α realising F. And so, as $\mathbf{sh}(\alpha)$ realises F, we are done by observing that $\mathbf{sh}(\alpha) \bullet g(*) = F(g)$. □

Hence we have shown under the assumption that type $\underline{2}$ is projective in $\mathcal{M}od(\mathcal{A}_{wb})$, p has a right-inverse and so $\underline{2}$ is a retract of $\underline{1}$ in $\mathcal{M}od(\mathcal{A})$. But we can show that this is impossible.

Proposition 5.5 *There is no right-inverse to p in $\mathcal{M}od(\mathcal{A})$.*

Proof: We assume that there is a morphism $e\colon N_\perp^{\mathcal{A}} \to \mathcal{A}$ such that $p \circ e = \mathrm{id}_{\underline{2}}$. A first observation is that from this assumption immediately follows that for all $F \in N_\perp^{\mathcal{A}}$ and all $f \in \mathcal{A}$, $F(f) \simeq e(F) \bullet f(*)$.

In the following we need the fact that the realisable functionals of $\mathcal{M}od(\mathcal{A})$ (what Longley calls *sequentially realisable functionals* in [Lon98b]) are exactly the *strongly stable* functionals.[6] From this and the fact that the stable and the pointwise order coincide on type $\underline{1}$, hence on $\mathcal{A}$, it is easy to show that

$$e(\Omega) \sqsubseteq e(F)$$

holds for all $F \in N_\perp^{\mathcal{A}}$, where $\Omega_{N_\perp^{\mathcal{A}}} := \Omega := \lambda f.\perp$.

With these prerequisites we are ready to show that under the above assumption

$$F = \Omega \iff e(F)(L*) \notin \{Ln \mid n \in \mathbb{N}\} \cup \{R_i m \mid i \in \mathbb{N}, m \in M\}$$

for all $F \in N_\perp^{\mathcal{A}}$.

Suppose $e(F)(L*)$ is not of the form Ln or $R_i m$ for some i, m or n. Let f be an element of $\mathcal{A}$. Then $F(f) \simeq e(F) \bullet f(*)$ diverges by the definition of $\bullet$. As f was arbitrary, $F = \Omega$.

[6] Strong stability is a notion due to Bucciarelli and Ehrhard (see [AC98] for an introduction).

Suppose $F = \Omega$ and $e(F)(L*) = Ln$ for some $n \in \mathbb{N}$. Then for all $f \in \mathcal{A}$, $F(f) \simeq e(F) \bullet f(*) = n$ in contradiction to $F(f) = \perp$.

On the other hand let $e(F)(L*) = R_i m$ for some $i \in \mathbb{N}, m \in M$. Let m' be any element of M not equal to m. We define $F_{m'}$ to be the element of $N_\perp^{\mathcal{A}}$ which is zero if its argument is defined at m', and is undefined otherwise. We define $f_{m'}$ to be the element of $\mathcal{A}$ which is only defined at m'. By the above remark we have $e(\Omega) \sqsubseteq e(F_{m'})$ and so $e(F_{m'})(L*) = R_i m$. But then we get $0 = F_{m'}(f_{m'}) \simeq e(F_{m'}) \bullet f_{m'}\uparrow$, which is a contradiction.

From this it follows that the functional

$$\begin{array}{rcl} T\colon N_\perp^{\mathcal{A}} & \to & N_\perp \\ F & \mapsto & \begin{cases} 0 & \text{if } e(F)(L*) = Ln \vee e(F)(L*) = R_i m \\ \perp & \text{if not} \end{cases} \end{array}$$

is realisable in $\mathcal{A}$ and therefore must be strongly stable.

But T is not stable as

$$\perp = T(\Omega) = T(F_1 \sqcap F_2) \neq T(F_1) \sqcap T(F_2) = 0 \sqcap 0 = 0$$

for any consistent pair, F_1 and F_2, of functionals different from $\Omega_{\underline{2}}$ and with $F_1 \sqcap F_2 = \Omega$. $\square$

We are now ready to state the main result:

Theorem 5.6 *Type $\underline{2}$ is not projective in $\mathcal{M}od(\mathcal{A}_{wb})$.*

Proof: If we assume that it is, Lemmata 5.1 to 5.4 apply. This is a contradiction to Proposition 5.5. $\square$

5.3 Conclusion

We can conclude from this result that the two combinatory algebras with divergence, $(\mathfrak{U}, D_{\mathfrak{U}})$ and $(\mathcal{A}_{wb}, D_{\mathcal{A}_{wb}})$, cannot be equivalent in the sense of Definition 2.10, because $\mathcal{M}od(\mathfrak{U})$ and $\mathcal{M}od(\mathcal{A}_{wb})$ would be equivalent as categories with dominances. However, this is a contradiction to the fact that $\underline{2}$ is projective in $\mathcal{M}od(\mathfrak{U})$ but not in $\mathcal{M}od(\mathcal{A}_{wb})$.

6 Further Work

We do know now that these two realisability models with their dominances are not equivalent. However I do not know if they are different by means of some kind of 'semantic' property. That is, a property which reflects how good model and programming language do fit. Such a property could be for instance the notion of *logical full abstractness*, an approach recently discussed by Longley [Lon99], in order to find finer grained utilities than fully abstractness and universality to distinguish between realisability models.

Logical full abstractness describes how realisability interpretations of logic formula coming either from the untyped universe of computation (the combinatory algebra) or from typed objects (the programming language), do agree.

In [Lon99, Section 4] is shown that essentially every realisability model coming from an universal object in a fully abstract model is logical full abstract, so we already know that $\mathcal{M}od(\mathfrak{U})$ is.

In my opinion, there is a good chance that $\mathcal{M}od(\mathcal{A}_{wb})$ is not, as objects not being projective can lead to failure of the axiom of choice in some instances. However, I was not able to proof this within this thesis. So further work might be to investigate this.

References

[AC98] R. M. Amadio and R.-L. Curien. *Domains and Lambda-Calculi*. Number 46 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1998.

[AJM94] S. Abramsky, R. Jagadeesan, and P. Malacaria. Full abstraction for PCF. *Lecture Notes in Computer Science*, 789:1–??, 1994.

[AL99] S. Abramsky and J. R. Longley. Realizability models based on history-free strategies. Unpublished draft, September 1999.

[HO95] J. M. E. Hyland and C.-H. L. Ong. Pi-calculus, dialogue games and PCF. In *7th Annual ACM Conference on Functional Programming Languages and Computer Architecture, La Jolla, California*, 1995.

[Kle45] S. C. Kleene. On the interpretation of intuitionistic number theory. *Journal of Symbolic Logic*, 10:109–124, 1945.

[Lon94] J. Longley. *Realizability Toposes and Language Semantics*. PhD thesis, University of Edinburgh, 1994.

[Lon98a] J. Longley. Realizability models for sequential computation. Unfinished draft, available from http://www.dcs.ed.ac.uk/home/jrl/pisa.ps.gz, September 1998.

[Lon98b] J. Longley. The sequentially realizable functionals. LFCS Report ECS-LFCS-98-402, University of Edinburgh, 1998.

[Lon99] J. Longley. Matching typed and untyped realizability. In L. Birkedal, J. van Oosten, G. Rosolini, and D.S. Scott, editors, *Workshop on Realizability Semantics and Applications*, volume 23 of *Electronic Notes in Computer Science*. Elsevier, 1999.

[LS97] J. R. Longley and A. E. Simpson. A uniform approach to domain theory in realizability models. *Mathematical Structures in Computer Science*, 7:469–506, 1997.

[Mar00] M. Marz. *A Fully Abstract Model for Sequential Computation*. PhD thesis, Technische Universität Darmstadt, 2000. To Appear.

[Mos90] P. D. Mosses. Denotational semantics. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics, chapter 11, pages 575–632. Elsevier Science Publishers B.V., 1990.

[MRS99] M. Marz, A. Rohr, and T. Streicher. Full abstraction and universality via realisability. In *Fourteenth Annual IEEE Symposium on Logic in Computer Science*, pages 174–182. IEEE Computer Society Press, 1999.

[Nic94] H. Nickau. Hereditarily sequential functionals. In *Proceedings of the Symposium on Logical Foundations of Computer Science: Logic at St. Petersburg*, Lecture Notes in Computer Science. Springer Verlag, 1994.

[Ong95] C.-H. L. Ong. Correspondence between operational and denotational semantics: the full abstraction problem for PCF. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 4, pages 269–356. cp, 1995.

[OR95] P. W. O'Hearn and J. G. Riecke. Kripke logical relations and PCF. *Information and Computation*, 120(1):107–116, 1995.

[Plo77] G. D. Plotkin. LCF considered as a programming language. *Theoretical Computer Science*, 5:223–255, 1977.

[Ros86] G. Rosolini. *Continuity and Effectiveness in Topoi.* PhD thesis, Oxford University, 1986.

[Sco69] D. S. Scott. A type theoretical alternative to ISWIM, CUCH, OWHY. Manuscript, University of Oxford, 1969.

[vO97] Jaap van Oosten. A Combinatory Algebra for Sequential Functionals of Finite Type. Technical Report 996, Department of Mathematics, Utrecht University, 1997.