CWI Tract 9

Models of the lambda calculus

C.P.J. Koymans

Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

1980 Mathematics Subject Classification: 03B40, 68F20
ISBN 90 6196 275 7

Printed in the Netherlands

ACKNOWLEDGEMENTS

The present text is an almost unaltered version of my Ph.D. thesis, written at the State University of Utrecht under guidance of Prof. D. van Dalen and Dr. H.P. Barendregt. I thank both of them for their teaching and support.

CONTENTS

INTRODUCTION

Although the theory of the pure, untyped lambda calculus as a formal system was rather well developed around 1940, it took the mathematical community about thirty years to produce the first models that were not of a syntactic nature. Before 1970 only the socalled term models existed, which were proven to be non-trivial by the Church-Rosser theorem for the lambda calculus, already published in 1936. The fact, that it took such a long time to define mathematical models calls for explanation. An evident problem was that the lambda calculus is formulated as an untyped theory, whereas the concept it tries to formalize, namely function application, is of a typed nature. (It is therefore no surprise that the typed variant of the pure lambda calculus has plenty of models that are easy to define.) But if we compare this situation with that of set theory, we conclude that although the $\in$-relation on sets has a typed nature, it was no problem to find models for the type free formulation of set theory as given by the Zermelo Fraenkel axioms. In fact set theory has as its intended model the universe of all sets as supplied by the cumulative hierarchy. Hence it was not only the type free aspect of the lambda calculus that caused trouble.

The intended interpretation of lambda terms is that of algorithms, that can themselves be considered as data and given as input to other algorithms. What seems reasonable at first sight is to identify the notion of algorithm with that of set theoretic function. In order to get the type free effect we would like to have $A^A \cong A$, where A is the domain of algorithms and A^A is the full set theoretic function space. But this is clearly impossible because of Cantor's theorem, if card(A) $>$ 1.

Another reason why lambda calculus does not fit in nicely with conventional systems is that it produces inconsistencies when combined with ordinary logic, viz. with negation. This was already experienced by Frege. He had a logical system, which essentially incorporated the full lambda calculus, see Aczel [1980]. This made the derivation of Russel's paradox, in the form of a fixed point of negation, possible.

The first solution to the problems as sketched above was given by Scott [1972]. He knew of it by the end of 1969. His idea was to identify an algorithm not with an arbitrary, but with a restricted kind of function, namely a continuous function. In order to do this he needed a category of topological spaces with continuous maps such that for any two spaces A,B a function space $[A \to B]$ could be formed, that had all important properties of ordinary set theoretic function spaces. A suitable category is that of complete lattices, where on each such complete lattice L an induced topology, the socalled Scott topology, can be defined. Call $U \subseteq L$ open if

(1) U is upwards closed, and

(2) If $\sup(D) \in U$ then there is a finite $D_0 \subseteq D$ such that $\sup(D_0) \in U$.

A function $f\colon L_1 \to L_2$ is continuous if and only if

$$f(\sup D) = \sup\{f(\sup D_0) \mid D_0 \subseteq D,\ D_0 \text{ finite}\}.$$

If L_1, L_2 are complete lattices, then so is $[L_1 \to L_2]$, being the lattice of all continuous functions from L_1 to L_2 with the pointwise ordering.
Now Scott started with an arbitrary complete lattice D_0 and defined inductively $D_{n+1} = [D_n \to D_n]$, $n \in \omega$. Clearly D_0 can be written as a retract of D_1:

$$\begin{tikzcd} D_0 \arrow[r, hook, shift left, "\varphi_0"] & D_1 \arrow[l, two heads, shift left, "\psi_0"] \end{tikzcd} \qquad \begin{array}{l} \varphi_0(a)(b) = a, \text{ for } a,b \in D_0, \\ \psi_0(f) = f(\bot), \text{ for } f \in [D_0 \to D_0]. \end{array}$$

This retract can be lifted to every level $\begin{tikzcd} D_n \arrow[r, hook, shift left, "\varphi_n"] & D_{n+1} \arrow[l, two heads, shift left, "\psi_n"] \end{tikzcd}$ by defining inductively

$$\begin{array}{ll} \varphi_{n+1}(f) = \varphi_n \circ f \circ \psi_n & , \text{ for } f \in [D_n \to D_n], \\ \psi_{n+1}(g) = \psi_n \circ g \circ \varphi_n & , \text{ for } g \in [D_{n+1} \to D_{n+1}]. \end{array}$$

Let D_∞ be the (projective) limit of the sequence

$$\begin{tikzcd} D_0 & D_1 \arrow[l, two heads, "\psi_0"'] & D_2 \arrow[l, two heads, "\psi_1"'] & D_3 \arrow[l, two heads, "\psi_2"'] & \ldots \arrow[l, two heads, "\psi_3"'] \end{tikzcd} .$$

Then it turns out that D_∞ is also the colimit (= direct limit) of the sequence

$$\begin{tikzcd} D_0 \arrow[r, hook, two heads, "\varphi_0"] & D_1 \arrow[r, hook, two heads, "\varphi_1"] & D_2 \arrow[r, hook, two heads, "\varphi_2"] & D_3 \arrow[r, hook, two heads, "\varphi_3"] & \ldots \end{tikzcd} .$$

and that $D_\infty \cong [D_\infty \to D_\infty]$.
Hence D_∞ induces in the obvious way a lambda calculus model, denoted by $\mathbb{D}_\infty$.

Once this first model was known, model theory of the lambda calculus developed rapidly. In Plotkin [1972] the graphmodel $\mathbb{P}\omega$, based on a set theoretic application, was introduced. This model was rediscovered by Scott and its usefulness shown by interpreting the lambda-calculus-based elementary programming language LAMBDA in it, see Scott [1976]. The graphmodel found its way into the semantics of programming languages, see Milne and Strachey [1976]. In the meantime Plotkin developed domains more suitable for interpretation of parallellism and nondeterminism in computations, see Plotkin [1976]. The cleanest and easiest set theoretical construction of a model was given in Engeler [1981]. A more detailed account of this important period can be found in appendix B of Scott [1976] or in his Turing award lecture, published as Scott [1977].

Meanwhile several people started to think about the general notion of a lambda calculus model. This turned out not to be so straightforward as might be expected. All "mathematical" models, constructed by Scott and Plotkin, interpreted lambda abstraction terms (that are terms of the form $\lambda x.M(x)$) by considering the function $d \mapsto M(\underline{d})$ and showing it to belong to a suitably defined function space $[X \to X]$, that can be embedded into X again. As we saw, because of cardinality reasons, if $\mathrm{card}(X) > 1$ then $[X \to X]$ cannot consist of all set theoretical functions from X to X. All models $\mathbb{D}_\infty$ and also the graphmodel resolve this difficulty by letting $[X \to X]$ consist of all continuous functions with respect to a suitable topology, the Scott topology. But this kind of interpretation has the following consequence for the models $\mathfrak{M}$ concerned: Every lambda abstraction term $(\lambda x.M)$ is completely determined by its applicative behaviour $d \mapsto (\lambda x.M)\underline{d} = M[x := \underline{d}]$. If this holds we say that $\mathfrak{M}$ is a *weakly extensional* model. We can express this by the following axiom:

$$\forall d \in |\mathfrak{M}| (\mathfrak{M} \models M(\underline{d}) = N(\underline{d})) \Rightarrow \mathfrak{M} \models \lambda x.M(x) = \lambda x.N(x).$$

This seems to be a reasonable assumption for a general lambda calculus model, were it not that the structure of all closed lambda terms (which certainly should be a model) does not satisfy the above axiom. This is an immediate consequence of the ω-incompleteness of the lambda calculus, as proved in Plotkin [1974].

Because of these facts the intuitive notion of model gave rise to two technical notions. The most comprehensive notion is that of a *lambda* *algebra*, that includes the closed term model mentioned above. The second notion is that of *lambda model*, being the subclass of the lambda algebras that satisfy the axiom of weak extensionality. In this text the unqualified term "model" will be used to denote the most general case, hence that of a lambda algebra.

Many people contributed to the general definition of a lambda calculus model. We refer the reader to Barendregt [1977],[1981], Berry [1980], [1981], Hindley and Longo [1980], Koymans [1979],[1982], Meyer [1980], [1982], Obtułowicz [1977],[1979], Obtułowicz and Wiweger [1978] and Scott [1980]. All of these definitions, except Berry's, are reviewed in Cooperstock [1981]. Any of the definitions, proposed in the papers above, belongs to one of the following three classes.

1. *Environment models* (see Hindley and Longo [1980], Koymans [1979] for lambda algebras and Meyer [1980] for lambda models).

   This approach is the most straightforward one, just postulating the existence of an interpretation map, satisfying the theory of the lambda calculus. In fact, an environment model is a structure $\mathfrak{M} = (X,\cdot)$ with for every assignment (sometimes called environment) ρ: variables → X a mapping

   $$[\![\cdot]\!]_\rho : \text{ lambda terms} \to X$$

   such that

   $$[\![x]\!]_\rho = \rho(x) \ ,$$
   $$[\![MN]\!]_\rho = [\![M]\!]_\rho \cdot [\![N]\!]_\rho \ .$$

   Moreover $\mathfrak{M}$ satisfies

   $$\lambda \vdash M = N \ \Rightarrow \ \forall\rho([\![M]\!]_\rho = [\![N]\!]_\rho).$$

   In this case $\mathfrak{M}$ is a lambda model if

   $$\forall d \in X([\![M]\!]_{\rho(x/d)} = [\![N]\!]_{\rho(x/d)}) \Rightarrow [\![\lambda x.M]\!]_\rho = [\![\lambda x.N]\!]_\rho.$$

   If $\mathfrak{M}$ is a lambda model it can be equivalently described as a functional domain in the terminology of Meyer [1980].

2. *First order models* (see Barendregt [1981] for lambda algebras and Meyer [1980] and Scott [1980] for lambda models).

At first sight lambda calculus is an ordinary equational theory, but in fact it is not. For the operation of lambda abstraction has the strength of a universal quantifier (assuming the axiom of weak extensionality), since

$$\lambda x.M = \lambda x.N \iff \forall x(M = N).$$

Moreover this axiom of weak extensionality does not hold in full generality, therefore the reduction as given above is not always possible. An adequate first-order definition can only be given after a purely equational theory is constructed, that is proof theoretically equivalent to lambda calculus. Such a theory is supplied by strong combinatory logic. Then a first-order model is a structure $\mathfrak{M} = (X,\cdot,s,k)$, such that $\mathfrak{M}$ satisfies the well-known axioms for the combinators S and K:

$$\mathfrak{M} \models sxyz = xz(yz) \wedge kxy = x.$$

Moreover $\mathfrak{M}$ has to satisfy an extra set A_β of five equations between terms in s,k , e.g.

$$\mathfrak{M} \models s(ks)(s(kk)) = s(kk)(s(s(ks)(s(kk)i))(ki)),$$

where $i = skk$.
The addition of the set of axioms A_β to combinatory logic makes this theory essentially equivalent to lambda calculus.
Now $\mathfrak{M}$ is a lambda model if

$$\mathfrak{M} \models \forall x(ax = bx) \rightarrow 1a = 1b\ ,$$

where $1 = s(ki)$.

3. *Categorical models* (see Berry [1981], Koymans [1982] for lambda algebras and Obtułowicz [1979] for lambda models; also see Meyers [1974] for the case of typed lambda calculus).

The most important advantage of this approach is that we can retain

the function interpretation of lambda abstraction terms without being forced to accept the axiom of weak extensionality. The reason is that category theory is an instrument that allows us to look at functions as transformations or algorithms with an intensional meaning.
The description of a lambda algebra now consists of a cartesian closed category C with a reflexive object U; reflexivity of U means that the function space U^U is a retract of U, notation $U^U \lhd U$.
Let

$$\begin{tikzcd} U^U \arrow[r, hook, "G"] & U \end{tikzcd} \quad , \quad \begin{tikzcd} U \arrow[r, two heads, "F"] & U^U \end{tikzcd}$$

be the corresponding section and projection respectively. The interpretation of the *untyped* lambda calculus is now possible since every element of U can be considered as a function by making use of the function-morphism (fun-operator) F and every function on U can be considered as an element of U by applying the graph-morphism (graph-operator) G.
The basic β-axiom of lambda calculus is satisfied because $F \circ G = id_{U^U}$.

In this case the structure defined is a lambda model if the category C has enough points in the following sense:

$$\forall f,g : A \to B (f \neq g \to \exists x : T \to A(f \circ x \neq g \circ x)) .$$

Here T is the terminal object in C.

For each of these three approaches it is easy to describe the characteristics of extensional models. Extensionality is the property that all elements of the domain (not only, as in the weakly extensional case, the lambda abstraction terms) are determined by their applicative behaviour. That is, $\mathfrak{M}$ is *extensional* if

$$\forall a,b \in |\mathfrak{M}| \ (\forall d \in |\mathfrak{M}| \ (ad = bd) \to a = b) .$$

All of the three ways of introducing lambda calculus models have their advantages and disadvantages. The environment models are simple to define and lead to the easiest approach to certain kinds of models like the term models and the filter models. For this last kind of model, see Barendregt, Coppo and Dezani-Ciancaglini [1983]. A disadvantage is that their definition depends on the notion of provability and hence is rather syntactical. The first-order structures have the advantage that they clearly indicate the model theoretic status of lambda calculus models and provide the connection with classical universal algebra. On the other hand it is often quite hard to show directly using the first-order axioms that a certain structure is a model. This is clearly recognizable for the socalled hypergraphmodel as defined in Sanchis [1979], which is the subject in chapter 4 of this book. The categorical models are important because they unify the notions of lambda algebra and lambda model and because they are the most natural structures for the lambda calculus from a mathematical point of view. For instance the models $\mathbb{D}_\infty$ and $\mathbb{P}\omega$ are most easily understood in the context of the category CLATT of all continuous lattices with Scott continuous mappings as morphisms. For a comprehensive treatment of continuous lattices we refer the reader to Gierz et al. [1980]. On the other hand the categorical approach introduces quite a lot of extra structure which in some cases is just unnecessary.

When we consider just models for combinatory logic the first two approaches, the environmental and first-order one, are clearly equivalent. There is also some work being done in the categorical direction in this case, see Longo and Moggi [1983]. It is hoped that this kind of categorical analysis will give a better understanding of the quite complicated axiom set A_β.

Some functorial aspects of the constructions, given in this book, interrelating the three approaches to model theory were analyzed by Adachi [1983] and Yokouchi [1983].

Overview.

The first chapter of the present text extensively reviews the basic definitions and properties of the environmental and first-order structures, with all useful variants. The central topic of (weak) extensionality is treated and moreover the correct notion of homomorphism is defined.

The third approach to lambda calculus structures, that of categorical models, is the subject of chapter 2. The idea of interpreting lambda calculus in a cartesian closed category is quite natural and goes back to Scott. The fact that every lambda algebra, as defined by the environmental or first-order approach, can be represented in this categorical context, cf. theorem 2.4.10. of this text, was first proved in Koymans [1981], published as Koymans [1982]. The construction, needed to prove this fact, makes use of a method of enlarging the class of objects in a category, quite similar to the method used in Karoubi [1978].

Chapter 3 is an illustration of how categorical techniques may be successfully applied to analyze an observation of Scott [1974], defining an extensional lambda calculus model, by a method resembling the construction of $\mathbb{D}_\infty$, but now inside another model (in Scott's article the graphmodel $\mathbb{P}\omega$). We introduce the notion of a derived lambda algebra and characterize the theory of this derived model by defining a translation on the set of lambda terms. This results in the determination of the theory of $\mathbb{P}_\infty$ (the model in Scott [1974]) and of all intermediate models $(\mathbb{P}_n)_{n\in\omega}$ used in this construction. It turns out that $\mathrm{Th}(\mathbb{P}_\infty) = \mathrm{Th}(\mathbb{D}_\infty)$, while all intermediate models $\mathbb{P}_n, n\in\omega$, are elementarily equivalent to $\mathbb{P}_0 = \mathbb{P}\omega$. The material of this chapter was first presented in Koymans [1983].

Chapter 4 of this book pays attention to the particular properties of the socalled hypergraphmodel of Sanchis [1979]. This structure has the same domain as the graphmodel, but its application is a Π^1_1-variant of ordinary graph application. Sanchis's original motivation was to study generalizations of certain kinds of recursion theoretic reducibilities like (hyper) enumeration and Turing reducibility. These generalizations can best be formulated in terms of application in Scott's graphmodel for the ordinary r.e. case and in the new hypergraphmodel for the hyperarithmetical case. In his article Sanchis shows that the hypergraphmodel is combinatory complete, hence that it is a model for combinatory logic, but the question whether a lambda calculus interpretation can be given remains unanswered. We will show that it is not possible for the hypergraphmodel to be considered as a lambda model by defining a suitable interpretation for abstraction. This in contrast with the ordinary graphmodel $\mathbb{P}\omega$. The results in this chapter were first presented in Koymans [1983A].

CHAPTER 1

LAMBDA CALCULUS MODELS: LAMBDA ALGEBRAS AND LAMBDA MODELS

1.1. *Syntactical aspects of lambda calculus model theory.*

Since the subject matter of the present text is the modeltheory of the *pure, untyped* lambda calculus, it is appropriate to look at the syntax of the lambda calculus (as a theory) first. It will be convenient to extend the expressive power of the language slightly, since we want to talk about the elements of the models we will be considering.

1.1.1. Definition (syntax of the lambda calculus).
Let C be a set of constants and Vars = $\{v_0,v_1,...\}$ a denumerably infinite set of variables; x,y,z,... range over Vars.

(i) The set $\Lambda(C)$ of *lambda terms* with constants from C is inductively defined by

$$\begin{array}{rcll} \text{Vars} & \subseteq & \Lambda(C) & \\ C & \subseteq & \Lambda(C) & \\ M,N \in \Lambda(C) & \Rightarrow & (MN) \in \Lambda(C) & \text{(application)} \\ M \in \Lambda(C) & \Rightarrow & (\lambda x.M) \in \Lambda(C) & \text{(abstraction).} \end{array}$$

(ii) We say that $\lambda x.$ binds all occurrences of x in $(\lambda x.M)$.
FV(M) is the set of free variables of M; these are all variables not bound by any $\lambda x.$ occurring in M.
M is *closed* if FV(M) = $\emptyset$.
M[x := N] denotes substitution of N for all free occurrences of x in M. As usual, a substitution is only legitimate if no free variables of N become bound by some abstraction in M after substitution. From now on we will assume that all substitutions are legitimate.
Both FV(M) and M[x := N] have a straightforward inductive defi-

nition.

(iii) The theory $\lambda(C)$ is the set of equations between lambda terms, generated by the following axioms and rules (where $M,N,L \in \Lambda(C)$)

$$\begin{array}{rcll} M & = & M & \\ \lambda x.M & = & \lambda y.M[x:=y] & (\alpha\text{-axiom}) \\ (\lambda x.M)N & = & M[x:=N] & (\beta\text{-axiom}) \\ M=N & \Rightarrow & N=M & \\ M=N,\ N=L & \Rightarrow & M=L & \\ M=N & \Rightarrow & ML=NL,\ LM=LN & \\ M=N & \Rightarrow & \lambda x.M = \lambda x.N & (\xi\text{-rule}). \end{array}$$

In the α-axiom we need the restriction that $y \notin FV(M)$. The so-called α-convertible terms (which are provably equal without the β-axiom) differ only in their bound variables; it is a standard convention that these terms are identified as will be done mostly in this book.

Let T be a set of equations between lambda terms.

(iv) $T \vdash M=N$ means that $M=N$ can be derived from the equations in T and the axioms and rules of (iii).

$\lambda(C) \vdash M=N$, or even $\vdash M=N$, is the same as $\emptyset \vdash M=N$.

(v) T is a $\lambda(C)$-*theory* if T is a set of closed equations such that

$$T \vdash M=N \;\Rightarrow\; (M=N) \in T$$

for all closed lambda terms M,N.

□

This definition is the usual one of the syntax of the lambda calculus, except that a set of constants is introduced. The only purpose of this is to be able, when considering models, to talk formally about the elements of these models in our lambda theories.

1.1.2. Notations. We adopt the usual notational conventions. Unessential brackets are left out.

Application is associated to the left:

$MN_1 \ldots N_n$ stands for $(\ldots(MN_1)\ldots N_n)$.

Abstraction is associated to the right:

$\lambda x_1 \ldots x_n.M$ stands for $(\lambda x_1 \ldots (\lambda x_n.M)\ldots)$.

$\Lambda^0(C) = \{M \in \Lambda(C) \mid FV(M) = \emptyset\}$ is the set of closed lambda terms.
$\Lambda^0(C,\vec{x}) = \{M \in \Lambda(C) \mid FV(M) \subseteq \{\vec{x}\}\}$.
We write $\lambda, \Lambda, \Lambda^0, \Lambda^0(\vec{x}), \ldots$ instead of $\lambda(\emptyset), \Lambda(\emptyset), \Lambda^0(\emptyset), \Lambda^0(\emptyset,\vec{x}), \ldots$.
If $\vec{x}$ is a sequence of distinct variables we have a notion of simultaneous substitution $M[\vec{x} := \vec{N}]$.

□

The first impression one has is that lambda calculus is a purely equational theory, but a closer look reveals that the variable binding term operator $\lambda x.$ has the internal strength of a quantifier. This will be seen more clearly when we discuss the notion of weak extensionality. Nevertheless, if one gives up weak extensionality, there exists an equivalent formulation of lambda calculus that is purely equational, see section 1.2.

The next definition will lead in a natural way to a notion of model for the lambda calculus. The disadvantage of this definition is that it is rather crude. The advantages are its naturalness and its usefulness when defining certain kinds of models, e.g. term models and filter models. Despite its naturality the definition is of a rather recent date, cf. Hindley and Longo [1980], Berry [1980] and Koymans [1979].

The following definition is used only temporarily.

1.1.3. Definition. A *proto lambda algebra* is a structure $\mathfrak{M} = (X, [\![\cdot]\!])$, where $[\![\cdot]\!] : \Lambda^0(\underline{X}) \to X$ and $\underline{X} = \{\underline{a} \mid a \in X\}$, such that

(a) $[\![\underline{a}]\!] = a$ for all $a \in X$,

(b) $Th(\mathfrak{M}) \vdash M = N \Rightarrow [\![M]\!] = [\![N]\!]$ for all $M, N \in \Lambda^0(\underline{X})$.

Here $Th(\mathfrak{M}) = \{M = N \mid M, N \in \Lambda^0(\underline{X}) \wedge [\![M]\!] = [\![N]\!]\}$.

(Hence (b) can be equivalently formulated as $Th(\mathfrak{M})$ is a $\lambda(\underline{X})$-theory.)

□

Note that in this definition, the domain of the function $[\![\cdot]\!]$ is a set of closed lambda terms. It is easy to extend this to all lambda terms, using assignments, in the following way.

1.1.4. Definition (satisfaction).

Let $\mathfrak{M} = (X, [\![\cdot]\!])$ be a structure with $[\![\cdot]\!] : \Lambda^0(\underline{X}) \to X$.

(i) An *assignment* in X is a mapping $\rho : \mathrm{Vars} \to X$.

Let Ass(X) be the set of assignments in X. If $\rho \in \mathrm{Ass}(X)$, $x \in \mathrm{Vars}$ and $d \in X$ then $\rho(x/d) \in \mathrm{Ass}(X)$ is defined by

$$\rho(x/d)(y) = \begin{cases} d & \text{, if } y \equiv x, \\ \rho(y), & \text{if } y \not\equiv x. \end{cases}$$

(ii) Define $\llbracket\cdot\rrbracket_{\bullet} : \Lambda(\underline{X}) \times \mathrm{Ass}(X) \to X$ by

$$\llbracket M(\vec{x})\rrbracket_{\rho} = \llbracket M[\vec{x} := \underline{\rho(\vec{x})}]\rrbracket .$$

We observe the convention that $M(\vec{x})$ is an informal notation for a term M with $FV(M) \subseteq \{\vec{x}\}$. In this case $M[\vec{x} := \underline{\vec{a}}]$ may be denoted as $M(\underline{\vec{a}})$.

(iii) $\mathfrak{M},\rho \models M = N \iff \llbracket M\rrbracket_{\rho} = \llbracket N\rrbracket_{\rho}$.

(iv) $\mathfrak{M},\rho \models \varphi$, for a first-order formula φ over the equations of the lambda calculus, is defined in the usual way.

(v) $\mathfrak{M} \models \varphi \iff \forall\rho \in \mathrm{Ass}(X)\,(\mathfrak{M},\rho \models \varphi)$.

□

Let us note a few easy consequences of this definition.

1.1.5. Lemma. Let $\mathfrak{M} = (X,\llbracket\cdot\rrbracket)$ be a proto lambda algebra, $M,N \in \Lambda(\underline{X})$ and $\rho,\sigma \in \mathrm{Ass}(X)$.

(i) If $M \in \Lambda^{0}(\underline{X})$ then $\llbracket M\rrbracket_{\rho} = \llbracket M\rrbracket$.

(ii) $\llbracket x\rrbracket_{\rho} = \rho(x)$.

(iii) If $\rho \restriction FV(M) = \sigma \restriction FV(M)$ then $\llbracket M\rrbracket_{\rho} = \llbracket M\rrbracket_{\sigma}$.

(iv) If $\mathrm{Th}(\mathfrak{M}) \vdash M = N$ then $\mathfrak{M} \models M = N$.

Proof:

(i),(ii) and (iii) are easy.

(iv) Note that, comparing with 1.1.3.(b), only the fact that M and N may be open is new. Now let $M,N \in \Lambda^{0}(\underline{X},\vec{x})$. Then

$$\begin{aligned} \mathrm{Th}(\mathfrak{M}) \vdash M = N &\Rightarrow \mathrm{Th}(\mathfrak{M}) \vdash \lambda\vec{x}.M = \lambda\vec{x}.N \\ &\Rightarrow \mathrm{Th}(\mathfrak{M}) \vdash (\lambda\vec{x}.M)\underline{\vec{a}} = (\lambda\vec{x}.N)\underline{\vec{a}} \quad \text{, for all } \vec{a} \in X, \\ &\Rightarrow \mathrm{Th}(\mathfrak{M}) \vdash M[\vec{x} := \underline{\vec{a}}] = N[\vec{x} := \underline{\vec{a}}], \text{ for all } \vec{a} \in X, \\ &\Rightarrow \mathfrak{M} \models M[\vec{x} := \underline{\vec{a}}] = N[\vec{x} := \underline{\vec{a}}] \quad \text{, for all } \vec{a} \in X, \\ &\Rightarrow \mathfrak{M},\rho \models M = N \quad \text{, for all } \rho \in \mathrm{Ass}(X), \\ &\Rightarrow \mathfrak{M} \models M = N. \end{aligned}$$

□

If one looks at definition 1.1.3. it seems to be too general. To show that this is not the case we will treat first-order logic in this way and see that this leads to the usual definition.

1.1.6. Intermezzo. Let L be an algebraic first-order language with function symbols $\{f_i \mid i \in I\}$ and constants $\{c_k \mid k \in K\}$. Now let $\mathfrak{M} = (X, \llbracket \cdot \rrbracket)$ with $\llbracket \cdot \rrbracket$ a mapping from the closed first-order terms, with constants from X, to X. Assume as before that $\llbracket \underline{a} \rrbracket = a$ for all $a \in X$ and $\mathrm{Th}(\mathfrak{M}) \vdash s = t \Rightarrow \mathfrak{M} \models s = t$ for closed terms s,t. Then define, if f is an n-ary function symbol, $f^{\mathfrak{M}} : X^n \to X$ by

$$f^{\mathfrak{M}}(a_1,\ldots,a_n) = \llbracket f(\underline{a_1},\ldots,\underline{a_n}) \rrbracket .$$

Define furthermore for a constant symbol c that $c^{\mathfrak{M}} = \llbracket c \rrbracket$.
Then we want to show that it follows that

$$\llbracket f(t_1,\ldots,t_n) \rrbracket = f^{\mathfrak{M}}(\llbracket t_1 \rrbracket,\ldots,\llbracket t_n \rrbracket)$$

for all closed terms $t_1,\ldots,t_n$.
Indeed, $\mathfrak{M} \models t_i = \underline{\llbracket t_i \rrbracket}$ for $1 \leqslant i \leqslant n$.
Hence $\mathrm{Th}(\mathfrak{M}) \vdash t_i = \underline{\llbracket t_i \rrbracket}$ for $1 \leqslant i \leqslant n$.
Therefore, by a well-known rule of equality, we get

$$\mathrm{Th}(\mathfrak{M}) \vdash f(t_1,\ldots,t_n) = f(\underline{\llbracket t_1 \rrbracket},\ldots,\underline{\llbracket t_n \rrbracket}).$$

Going back to $\mathfrak{M}$ again we find that

$$\mathfrak{M} \models f(t_1,\ldots,t_n) = f(\underline{\llbracket t_1 \rrbracket},\ldots,\underline{\llbracket t_n \rrbracket}).$$

That is $\llbracket f(t_1,\ldots,t_n) \rrbracket = \llbracket f(\underline{\llbracket t_1 \rrbracket},\ldots,\underline{\llbracket t_n \rrbracket}) \rrbracket$.

Applying the definition of $f^{\mathfrak{M}}$ to the right-hand-side of this equality gives us the desired result. The upshot of all this is that the whole interpretation of terms is determined by $\{f_i^{\mathfrak{M}} \mid i \in I\}$ and $\{c_k^{\mathfrak{M}} \mid k \in K\}$ and that this interpretation can be defined inductively as usual.

□

Using the ideas of this intermezzo it is possible to make definition 1.1.3. a little less crude by noting that application is just a binary operation.

1.1.7. Definition. Let $\mathfrak{M} = (X, [\![\cdot]\!])$ be a proto lambda algebra. A binary operation $\cdot : X^2 \to X$ is defined by

$$a \cdot b = [\![\underline{a}\ \underline{b}]\!] \qquad (\textit{application}).$$

□

1.1.8. Fact. Let $\mathfrak{M} = (X, [\![\cdot]\!])$ be a proto lambda algebra and $\cdot : X^2 \to X$ the corresponding binary operation.
Then for all $M, N \in \Lambda^0(\underline{X})$ we have

$$[\![MN]\!] = [\![M]\!] \cdot [\![N]\!].$$

Proof: Cf. the corresponding proof for the first-order case in intermezzo 1.1.6.

□

The only situation where we really made use of the strength of $\mathrm{Th}(\mathfrak{M})$ is in the proof of this fact. On the other hand, adding the binary operation to the definition of a proto lambda algebra eliminates the need for $\mathrm{Th}(\mathfrak{M})$, which is the content of the next lemma.

1.1.9. Lemma. Let $\mathfrak{M} = (X, \cdot, [\![\cdot]\!])$ be a structure with $\cdot : X^2 \to X$ and $[\![\cdot]\!] : \Lambda^0(\underline{X}) \to X$, such that for all $a \in X$ and $M, N \in \Lambda^0(\underline{X})$

(a) $[\![\underline{a}]\!] = a$,
(b) $[\![MN]\!] = [\![M]\!] \cdot [\![N]\!]$,
(c) $\lambda(\underline{X}) \vdash M = N \Rightarrow \mathfrak{M} \models M = N$.

Then for all $M, N \in \Lambda^0(\underline{X})$

$$\mathrm{Th}(\mathfrak{M}) \vdash M = N \Rightarrow \mathfrak{M} \models M = N.$$

Hence $(X, [\![\cdot]\!])$ is a proto lambda algebra and $\cdot$ is the binary operation as given in definition 1.1.7.

Proof: Let us show that the following stronger property holds: for all $M, N \in \Lambda^0(\underline{X}, \vec{x})$

$$\mathrm{Th}(\mathfrak{M}) \vdash M = N \Rightarrow \mathfrak{M} \models \lambda\vec{x}.M = \lambda\vec{x}.N.$$

This is shown by induction on the length of proof of $M = N$ from $\mathrm{Th}(\mathfrak{M})$. The lambda calculus axioms are treated by (c). If

$(M=N) \in \mathrm{Th}(\mathfrak{M})$ then $M,N \in \Lambda^0(\underline{X})$ and $\lambda(\underline{X}) \vdash \lambda\vec{x}.M = (\lambda y\vec{x}.y)M$ and similarly for N. Hence by (c),(b) $\mathfrak{M} \models \lambda\vec{x}.M = \lambda\vec{x}.N$. Note that in the case of the ξ-rule everything is automatically clear by the way the above property is stated. Let me do finally one other rule. Assume $M=N \Rightarrow ML=NL$ is the last rule applied. Then by induction hypothesis $\mathfrak{M} \models \lambda\vec{x}.M = \lambda\vec{x}.N$, where $\vec{x}$ is such that $M,N,L \in \Lambda^0(\underline{X},\vec{x})$. We have to show that $\mathfrak{M} \models \lambda\vec{x}.ML = \lambda\vec{x}.NL$. But we notice that $\lambda(\underline{X}) \vdash \lambda\vec{x}.ML = (\lambda y\vec{x}.y\vec{x}L)(\lambda\vec{x}.M)$ and similarly for N. Hence the result follows by some applications of (c) and (b).
The last part of the lemma follows easily.

□

Because of this lemma the final version of the definition of a proto lambda algebra takes the following form.

1.1.10. Definition. A *proto lambda algebra* is a structure $\mathfrak{M} = (X, \cdot, [\![\cdot]\!])$, where $\cdot : X^2 \to X$ and $[\![\cdot]\!] : \Lambda^0(\underline{X}) \to X$ are such that for all $a \in X$ and $M,N \in \Lambda^0(\underline{X})$

(a) $[\![\underline{a}]\!] = a$,

(b) $[\![MN]\!] = [\![M]\!] \cdot [\![N]\!]$,

(c) $\lambda(\underline{X}) \vdash M=N \Rightarrow \mathfrak{M} \models M=N$.

□

Now that we made the role of application more explicit, one would hope that something similar could be done for abstraction. The first thing that comes to mind is to extend lambda calculus to a first-order theory, while interpreting the ξ-rule as being equivalent to the axiom

$$\forall x(M=N) \to \lambda x.M = \lambda x.N \qquad (\xi).$$

Because of the β-axiom we really have an equivalence here. The ξ-axiom, also called the axiom of weak extensionality, states that functional lambda terms (these are lambda abstraction terms) are determined by their applicative behaviour. But this is not generally valid. Instead, every lambda term should be considered as a program that has its own intensional character apart from its input-output behaviour. Perhaps the strongest evidence for this is the following proposition due to Plotkin.

1.1.11. Proposition (Plotkin terms).

There exist terms $F,G \in \Lambda^0$ such that

(a) for all $Z \in \Lambda^0$ we have $\lambda \vdash FZ = GZ$,

(b) $\lambda \nvdash Fx = Gx$ for a variable x.

Proof: See Barendregt [1981], ch. 17 §3.

□

This proposition tells us that the ξ-axiom does not hold in the so-called closed term model. Now we will introduce these term models formally. Together with the filter models they show how easily the crude interpretation of definition 1.1.10. works.

1.1.12. Definition (open term models).

Let T be a $\lambda(C)$-theory.

Let $=_T$ be the equivalence relation on $\Lambda(C)$ defined by

$$M =_T N \iff T \vdash M = N.$$

$[M]_T$ denotes the equivalence class of M.

Let $X = \Lambda(C)/=_T$ and define $\cdot : X^2 \to X$ by

$$[M]_T \cdot [N]_T = [MN]_T.$$

(This does not depend on the choice of representative.)

Furthermore for any $M \in \Lambda^0(\underline{X})$ let

$$[\![M]\!] = [M']_T,$$

where M' is obtained from M by replacing every constant of the form $\underline{[N]_T}$ by N (if necessary modulo some renaming in M to avoid binding free variables of N). Note again that this does not depend on the choice of representative of $[N]_T$. Finally the *open term model* of T is the structure $\mathfrak{M}(T) = (X,\cdot,[\![\cdot]\!])$.

□

1.1.13. Lemma. Let T be a $\lambda(C)$-theory. Then $\mathfrak{M}(T)$ is a proto lambda algebra.

Proof: We check the conditions in definition 1.1.10.

(a) $[\![\underline{[N]_T}]\!] = [N]_T$ by definition.

(b) $\llbracket MN\rrbracket = [M'N']_T = [M']_T \cdot [N']_T = \llbracket M\rrbracket \cdot \llbracket N\rrbracket$.

(c) Let $M,N \in \Lambda^0(|\mathfrak{M}(T)|)$ and suppose that $\lambda(\mathfrak{M}(T)) \vdash M = N$. Choose $M_1, N_1 \in \Lambda^0(\vec{x})$ and $\vec{L} \in \Lambda(C)$ such that $M \equiv M_1[\vec{x} := \underline{[\vec{L}]_T}]$ and $N \equiv N_1[\vec{x} := \underline{[\vec{L}]_T}]$. Then clearly $\lambda \vdash M_1 = N_1$ and hence $T \vdash M_1[\vec{x} := \vec{L}] = N_1[\vec{x} := \vec{L}]$.

Therefore,

$$\begin{aligned} \llbracket M\rrbracket &= \llbracket M_1[\vec{x} := \underline{[\vec{L}]_T}]\rrbracket \\ &= [M_1[\vec{x} := \vec{L}]]_T \\ &= [N_1[\vec{x} := \vec{L}]]_T \\ &= \llbracket N_1[\vec{x} := \underline{[\vec{L}]_T}]\rrbracket \\ &= \llbracket N\rrbracket . \end{aligned}$$

Hence $\mathfrak{M}(T) \models M = N$, what had to be shown. □

Since the Plotkin terms talk about closed terms, we should go on by defining the socalled closed term models. But there exists a general procedure to define "closed" submodels of given models.

1.1.14. <u>Definition</u> (interiors).

Let $\mathfrak{M} = (X, \cdot, \llbracket \cdot \rrbracket)$ be a proto lambda algebra. Define $X^0 = \{\llbracket M\rrbracket \mid M \in \Lambda^0\}$ and $\llbracket M\rrbracket^0 = \llbracket M\rrbracket$ for any $M \in \Lambda^0(\underline{X^0})$. The *interior* of $\mathfrak{M}$ is the structure $\mathfrak{M}^0 = (X^0, \cdot, \llbracket \cdot \rrbracket^0)$. □

1.1.15. <u>Lemma</u>. The interior of a proto lambda algebra $\mathfrak{M}$ is well-defined and again a proto lambda algebra.

Proof: That $\mathfrak{M}^0$ is a proto lambda algebra, once it is well-defined, follows easily from definition 1.1.10. To show well-definedness we have to check first of all that X^0 is closed under $\cdot$. But $\llbracket M\rrbracket \cdot \llbracket N\rrbracket = \llbracket MN\rrbracket \in X^0$, whenever $M,N \in \Lambda^0$. Furthermore we want to show that $\llbracket M\rrbracket^0 \in X^0$ for any $M \in \Lambda^0(\underline{X^0})$. Let $M_1 \in \Lambda^0(\vec{x})$, $\vec{N} \in \Lambda^0$ such that $M \equiv M_1[\vec{x} := \underline{\llbracket \vec{N}\rrbracket}]$. Then

$$\begin{aligned} \llbracket M\rrbracket^0 &= \llbracket M_1[\vec{x} := \underline{\llbracket \vec{N}\rrbracket}]\rrbracket \\ &= \llbracket M_1[\vec{x} := \vec{N}]\rrbracket \in X^0 . \end{aligned}$$

□

1.1.16. Definition (closed term models).

Let T be a $\lambda(C)$-theory. The *closed term model* $\mathfrak{M}^0(T)$ is the interior of the open term model $\mathfrak{M}(T)$.

□

Equivalently, a definition like 1.1.12. could be given, restricting all terms to just closed terms. (Strictly speaking this would only give an isomorphic copy.) Let us now state formally the earlier introduced and very important concept of weak extensionality.

1.1.17. Definition. Let $\mathfrak{M}$ be a proto lambda algebra.

(i) $\mathfrak{M}$ is *weakly extensional* if $\mathfrak{M} \models \xi$, where ξ is the following axiomscheme:

$$\forall x(M = N) \rightarrow \lambda x.M = \lambda x.N,$$

where $M,N \in \Lambda^0(\mathfrak{M},x)$.

(ii) A *proto lambda model* is a weakly extensional proto lambda algebra.

□

1.1.18. Corollary (to the existence of Plotkin terms in 1.1.11.).

$\mathfrak{M}^0(\lambda) \not\models \xi$, that is $\mathfrak{M}^0(\lambda)$ is a proto lambda algebra, but not a proto lambda model.

Proof: Let F,G be the Plotkin terms in proposition 1.1.11. Take $M = Fx$, $N = Gx$. Then for all $Z \in \Lambda^0$

$$\begin{aligned} [\![F[\underline{Z}]]\!] &= [FZ] \\ &= [GZ] \quad \text{, by proposition 1.1.11.,} \\ &= [\![G[\underline{Z}]]\!]. \end{aligned}$$

Hence $\mathfrak{M}^0(\lambda) \models \forall x(M = N)$.

But on the other hand

$$\begin{aligned} \mathfrak{M}^0(\lambda) \models \lambda x.M = \lambda x.N &\Longleftrightarrow [\![\lambda x.Fx]\!] = [\![\lambda x.Gx]\!] \\ &\Longleftrightarrow \lambda \vdash \lambda x.Fx = \lambda x.Gx \\ &\Longleftrightarrow \lambda \vdash Fx = Gx. \end{aligned}$$

Hence by proposition 1.1.11.(b) $\mathfrak{M}^0(\lambda) \not\models \lambda x.M = \lambda x.N$.

We conclude $\mathfrak{M}^0(\lambda) \not\models \xi$.

□

On the other hand we can extend lemma 1.1.13. as follows.

1.1.19. Proposition. Let T be a $\lambda(C)$-theory, then

(i) $\mathfrak{M}(T)$ is a proto lambda model.

(ii) $\mathfrak{M}^0(T)$ is a proto lambda algebra.

Proof:

(i) By lemma 1.1.13. $\mathfrak{M}(T)$ is a proto lambda algebra. It remains to show that $\mathfrak{M}(T) \models \xi$. Hence let $M,N \in \Lambda^0(\mathfrak{M}(T), x)$ and suppose

$$\mathfrak{M}(T) \models \forall x (M = N).$$

This implies that

$$\mathfrak{M}(T) \models M[x := \underline{[x]_T}] = N[x := \underline{[x]_T}].$$

By definition of interpretation in $\mathfrak{M}(T)$ we get $[M']_T = [N']_T$. Therefore $[\lambda x.M']_T = [\lambda x.N']_T$ and again by definition of interpretation

$$\mathfrak{M}(T) \models \lambda x.M = \lambda x.N.$$

(ii) Apply lemma 1.1.15.

□

The combination of 1.1.18. and 1.1.19. shows us that $\mathfrak{M}(\lambda)$ is an example of a proto lambda model, whose interior is not a proto lambda model.

For another kind of model that is easy to define using the notion of proto lambda algebra, we refer the reader to the socalled filter models, see Barendregt, Coppo, Dezani-Ciancaglini [1983].

In case we have axiom ξ at our disposal it is possible to simplify the definition of a proto lambda algebra, which is then in fact automatically a proto lambda model. This is so, because axiom ξ makes proofs by induction on length of proof in the lambda calculus possible.

1.1.20. Proposition. Let $\mathfrak{M} = (X, \cdot, [\![\cdot]\!])$ be a structure with $\cdot : X^2 \to X$, $[\![\cdot]\!] : \Lambda^0(\underline{X}) \to X$ satisfying

(a) $[\![\underline{a}]\!] = a$ for all $a \in X$,

(b) $[\![MN]\!] = [\![M]\!] \cdot [\![N]\!]$ for all $M,N \in \Lambda^0(\underline{X})$,

(c) $[\![\lambda x.M(x)]\!]\cdot a = [\![M[x := \underline{a}]]\!]$ for all $M \in \Lambda^0(\underline{X},x)$ and $a \in X$,

(d) $\mathfrak{M} \models \xi$.

Then $\mathfrak{M}$ is a proto lambda model.

Proof: To verify condition 1.1.10.(c) we use induction on the length of proof of $M = N$ in $\lambda(\underline{X})$ to show

$$\lambda(\underline{X}) \vdash M = N \;\Rightarrow\; \mathfrak{M} \models M = N$$

for any $M,N \in \Lambda(\underline{X})$.
For the ξ-rule we use assumption (d).
For the β-axiom we use assumption (c), together with the fact that $[\![M[x := N]]\!] = [\![M[x := \underline{[\![N]\!]}]]\!]$ for all $M,N \in \Lambda^0(\underline{X})$, which is easily proved by induction on M, again making use of (d).

□

The approach towards semantics as followed in this section will henceforth be referred to as the crude approach.

1.2. *The approach of combinatory logic.*

In section 1.1. we tried to handle the semantics of the lambda calculus in a direct correspondence with its syntax, hence including the variable binding term operator λx. This last operator gave some trouble in that its interpretation need not always satisfy the very natural axiom of weak extensionality, see corollary 1.1.18. The purpose of this section is to look at this problem from another side by eliminating the use of λx. in favour of some new constants, the socalled combinators. The method for this elimination goes back to some ideas of Schönfinkel [1924], see Curry and Feys [1958]. We then arrive at the (at the syntactic level) completely equivalent theory of strong combinatory logic, that has the advantage of being equational in the pure sense. This means that all known modeltheory can be applied to this situation.

1.2.1. Definition (basic combinators of the lambda calculus).

(i) $S_\lambda = \lambda xyz.xz(yz)$.

(ii) $K_\lambda = \lambda xy.x$.

□

The basic fact that opens the way for carrying out the above sketched program is the following fact.

1.2.2. Fact. For any lambda term $M \in \Lambda(C)$ with $FV(M) = \{\vec{x}\}$ there exists another term N built up from $\vec{x}$, the constants from C occurring in M, S_λ and K_λ using only the operation of application, such that $\lambda(C) \vdash M = N$.

Proof: will be given in 1.2.10. □

Fact 1.2.2. suggests the following formal system.

1.2.3. Definition (syntax and rules of combinatory logic).
Let C be a set of constants.

(i) The set $C\ell(C)$ of combinatory terms with constants from C is inductively defined by:

$$\begin{array}{rcll} \text{Vars} & \subseteq & C\ell(C) & \\ C & \subseteq & C\ell(C) & \\ S,K & \in & C\ell(C) & \textit{(basic combinators)} \\ P,Q \in C\ell(C) & \Rightarrow & (PQ) \in C\ell(C). & \end{array}$$

(ii) CL(C) is the theory defined by the following axioms and rules, where $P,Q,R \in C\ell(C)$:

$$\begin{array}{rcll} P & = & P & \\ SPQR & = & PR(QR) & \text{(S-axiom)} \\ KPQ & = & P & \text{(K-axiom)} \\ P = Q & \Rightarrow & Q = P & \\ P = Q,\ Q = R & \Rightarrow & P = R & \\ P = Q & \Rightarrow & PR = QR,\ RP = RQ. & \end{array}$$

(In fact the first axiom is redundant.)

Let T be a set of equations between combinatory terms.

(iii) $T \vdash P = Q$ means that $P = Q$ can be derived from the equations in T and the axioms and rules of (ii).
$CL(C) \vdash P = Q$, or even $\vdash P = Q$, is the same as $\emptyset \vdash P = Q$.

(iv) T is a CL(C)-theory if T is a set of closed equations, closed under derivation. □

We observe that all occurrences of variables in combinatory terms are free and that the notion of (simultaneous) substitution can be defined in the obvious way.

1.2.4. Conventions. As always unessential brackets are left out and application is associated to the left.

$\mathcal{C}\ell^0(C) = \{P \mid FV(P) = \emptyset\}$ is the set of closed combinatory terms.

$\mathcal{C}\ell^0(C)(\vec{x}) = \mathcal{C}\ell^0(C,\vec{x}) = \{P \mid FV(P) \subseteq \{\vec{x}\}\}$.

□

The strength of the system of combinatory logic is that the operation of abstraction $\lambda x.$ in the lambda calculus can be simulated in CL, using only the two basic combinators S and K.

1.2.5. Definition (*abstraction* in CL).

(i) For every variable x a map $<x> : \mathcal{C}\ell(C) \to \mathcal{C}\ell(C)$ is defined inductively by

$$\begin{aligned} <x>x &= SKK \\ <x>P &= KP \qquad\qquad , \text{ if } x \notin FV(P)\ , \\ <x>(PQ) &= S(<x>P)(<x>Q), \text{ if } x \in FV(PQ). \end{aligned}$$

(ii) If $\vec{x} = x_1,\ldots,x_n$ then $<\vec{x}>P = (<x_1>\ldots(<x_n>P)\ldots)$.

□

Notice that abstraction in CL behaves very well with respect to substitution, that is

$$(<x>P)[y := Q] \equiv <x>(P[y := Q]).$$

1.2.6. Lemma. Let $P \in \mathcal{C}\ell(C)$, then

(i) $FV(<x>P) = FV(P)\backslash\{x\}$.

(ii) $\vdash (<x>P)x = P$.

Proof:

(i) Obvious.

(ii) By induction on P.

(a) $\vdash (<x>x)x = SKKx = Kx(Kx) = x$.

(b) If $x \notin FV(P)$, then

$\vdash (<x>P)x = KPx = P$.

(c) If $x \in FV(PQ)$, then
$\vdash (\langle x\rangle(PQ))x = S(\langle x\rangle P)(\langle x\rangle Q)x = (\langle x\rangle P)x((\langle x\rangle Q)x) = PQ$,
by the induction hypothesis.

□

The fact that abstraction can be simulated in combinatory logic gives us the possibility of the following translations.

1.2.7. Definition (*standard translations*).

(i) $\lambda : \mathcal{CL}(C) \to \Lambda(C)$ is defined inductively by:

$$\begin{aligned} \lambda(x) &= x &&, \text{ for } x \text{ a variable},\\ \lambda(c) &= c &&, \text{ for } c \in C,\\ \lambda(S) &= S_\lambda \\ \lambda(K) &= K_\lambda \\ \lambda(PQ) &= \lambda(P)\lambda(Q). \end{aligned}$$

(ii) $CL: \Lambda(C) \to \mathcal{CL}(C)$ is defined inductively by:

$$\begin{aligned} CL(x) &= x &&, \text{ for } x \text{ a variable},\\ CL(c) &= c &&, \text{ for } c \in C,\\ CL(MN) &= CL(M)CL(N)\\ CL(\lambda x.M) &= \langle x\rangle CL(M). \end{aligned}$$

□

As notations we use M_{CL} for CL(M) and P_λ for $\lambda(P)$.
Notice that this notation is consistent with the one introduced in definition 1.2.1.

1.2.8. Lemma. For all $P \in \mathcal{CL}(C)$, $M \in \Lambda(C)$ we have $FV(P) = FV(P_\lambda)$, $FV(M) = FV(M_{CL})$.

Proof: Induction on the structure of P, respectively M, using lemma 1.2.6.(i).

□

1.2.9. Lemma. Let $P,Q \in \mathcal{CL}(C)$, $M \in \Lambda(C)$.

(i) $\lambda(C) \vdash (\langle x\rangle P)_\lambda = \lambda x.P_\lambda$.

(ii) $CL(C) \vdash P = Q \Rightarrow \lambda(C) \vdash P_\lambda = Q_\lambda$.

(iii) $\lambda(C) \vdash M = (M_{CL})_\lambda$.

Proof:

(i) By induction on the structure of P. The most difficult case is $P \equiv QR$, where $x \in FV(P)$:

$$\lambda \vdash (\langle x\rangle(QR))_\lambda = S_\lambda(\langle x\rangle Q)_\lambda\,(\langle x\rangle R)_\lambda = S_\lambda(\lambda x.Q_\lambda)(\lambda x.R_\lambda) = \lambda x.Q_\lambda R_\lambda.$$

(ii) Easy induction on the proof of $P = Q$ in CL(C).

(iii) Induction on the structure of M, using (i).

□

1.2.10. Proof of 1.2.2. This is an immediate corollary of 1.2.9.(iii). It is clear that M and $(M_{CL})_\lambda$ contain the same constants from C.

□

From fact 1.2.2. and lemma 1.2.9.(ii) we can conclude that every lambda term can be represented as a combinatory term in such a way that combinatory logic does not prove anything, not provable in lambda calculus. But does it prove enough? The answer to this is "no!".

1.2.11. Lemma. There exist $M,N \in \Lambda^0$ such that $\lambda \vdash M = N$, but $CL \nvdash M_{CL} = N_{CL}$. So a fortiori 1.2.9.(ii) $\Leftarrow$ does not hold.

Proof: Take $M = \lambda x.x$ and $N = \lambda x.(\lambda y.y)x$.
Then $M_{CL} = SKK$ and $N_{CL} = S(K(SKK))(SKK)$. However, $CL \nvdash M_{CL} = N_{CL}$ (this follows from the familiar Church-Rosser theorem for CL, see Barendregt [1981], 7.2.4.).
Taking $P = M_{CL}$, $Q = N_{CL}$ one sees, using 1.2.9.(iii) that 1.2.9.(ii) $\Leftarrow$ doesn't hold.

□

In order to bring the theory of combinatory logic closer in strength to the lambda calculus, we give the following definition.

1.2.12. Definition (strong combinatory logic).

$$CL_\beta(C) = \{P = Q \mid P,Q \in \mathcal{C}\ell(C),\ \lambda(C) \vdash P_\lambda = Q_\lambda\}.$$

□

1.2.13. Proposition. $CL_\beta(C)$ is a CL(C)-theory with the following properties.

(i) $CL_\beta(C) \vdash P = Q \iff \lambda(C) \vdash P_\lambda = Q_\lambda$ for all $P,Q \in \mathcal{C}\ell(C)$.

(ii) $\lambda(C) \vdash M = N \iff CL_\beta(C) \vdash M_{CL} = N_{CL}$ for all $M,N \in \Lambda(C)$.

(iii) $CL_\beta(C) \vdash P = (P_\lambda)_{CL}$ for all $P \in \mathcal{C}\ell(C)$.

Proof:

(i) ⇒ is clear by induction on the length of proof of $P = Q$.

⇐ holds by definition.

In fact (i) can be read as saying that $CL_\beta(C)$ is a $CL(C)$-theory, if one restricts $CL_\beta(C)$ to closed equations.

(ii), (iii) follow from (i), using 1.2.9.(iii).

□

The theory $CL_\beta(C)$ can be finitely axiomatized over the usual axiom-schemes and rules of $CL(C)$. The method is due to Curry.

1.2.14. Definition.

Let A_β consist of the following five axioms:

(A1) $K = S(S(KS)(S(KK)K))(K(SKK))$

(A2) $S = S(S(KS)(S(K(S(KS)))(S(K(S(KK)))S)))(K(K(SKK)))$

(A3) $S(KK) = S(S(KS)(S(KK)(S(KS)K)))(KK)$

(A4) $S(KS)(S(KK)) = S(KK)(S(S(KS)(S(KK)(SKK)))(K(SKK)))$

(A5) $S(K(S(KS)))(S(KS)(S(KS))) =$
$S(S(KS)(S(KK)(S(KS)(S(K(S(KS)))S))))(KS)$.

□

1.2.15. Proposition.

$CL_\beta(C)$ can be axiomatized over $CL(C)$ by the set of axioms A_β, that is

$$CL_\beta(C) = \{P = Q \mid CL(C) + A_\beta \vdash P = Q\}.$$

Proof: See Barendregt [1981], chapter 7.3., where the history of the set of equations A_β can be traced back.

□

Needless to say that one look at definition 1.2.14. immediately reveals the modest practical usefulness of the set of axioms A_β. This in contrast with its theoretical importance. If one looks at the proof of proposition 1.2.15. one gets a better understanding of their meaning. A clear insight in these axioms and their complexity is however still lacking.

Now that lambda calculus has been "translated" into an equational theory one can approach the semantics along these lines, as will be done in the remainder of section 1.2.

1.2.16. Definition. An *applicative structure* is a structure $\mathfrak{M} = (X,\cdot)$ where $\cdot : X^2 \to X$ is a binary operation.

□

Although formally an applicative structure is nothing more than a nonempty set with a binary operation, this terminology has been chosen since in lambda calculus models this operation is intended to be application of functions to arguments.

We note that the language belonging to an applicative structure is just the language of combinatory logic without the two basic combinators S and K.

1.2.17. Definition. Let $\mathfrak{M} = (X,\cdot)$ be an applicative structure.

(i) The set of *terms over* $\mathfrak{M}$, that is $\mathfrak{T}(\mathfrak{M})$, is defined inductively by

$$\begin{aligned} \mathrm{Vars} &\subseteq \mathfrak{T}(\mathfrak{M}) \\ \underline{X} &\subseteq \mathfrak{T}(\mathfrak{M}) \\ P,Q \in \mathfrak{T}(\mathfrak{M}) &\Rightarrow (PQ) \in \mathfrak{T}(\mathfrak{M}). \end{aligned}$$

(Hence $\mathfrak{T}(\mathfrak{M}) = \{P \in \mathcal{C\ell}(\underline{X}) \mid S,K \text{ do not occur in } P\}$.)

(ii) $[\![\cdot]\!]_\circ : \mathfrak{T}(\mathfrak{M}) \times \mathrm{Ass}(X) \to X$ is defined inductively as usual by

$$\begin{aligned} [\![x]\!]_\rho &= \rho(x) \qquad , \text{for } x \in \mathrm{Vars}, \\ [\![\underline{a}]\!]_\rho &= a \\ [\![PQ]\!]_\rho &= [\![P]\!]_\rho [\![Q]\!]_\rho \,. \end{aligned}$$

(iii) Satisfaction is defined as usual, cf. 1.1.4.(iii), (iv), (v).

□

We observe the notational conventions of 1.2.4.

As such, applicative structures are not of interest for the subject of this text. For instance groups and other algebraic structures give examples of applicative structures. The important property that distinguishes the applicative structures that are useful for our purposes from others is the property of combinatory completeness, which says that any series of applications in whatever order can be represented in a canonical form.

1.2.18. Definition. Let $\mathfrak{M} = (X,\cdot)$ be an applicative structure. $\mathfrak{M}$ is called *combinatory complete* if for every $P \in \mathfrak{T}(\mathfrak{M})$ with free variables among $\vec{x}$ we have

$$\mathfrak{M} \models \exists y \forall \vec{x}(y\vec{x} = P).$$ □

In order to state this property in another way we present the following definition.

1.2.19. Definition. Let $\mathfrak{M} = (X,\cdot)$ be an applicative structure and $\varphi : X^n \to X$ a mapping.

(i) φ is *representable over* $\mathfrak{M}$ if $\exists y \in X\ \forall \vec{x} \in X\ (y\vec{x} = \varphi(\vec{x}))$.

(ii) φ is *algebraic over* $\mathfrak{M}$ if it is definable in $\mathfrak{M}$, that is if there exists a term $P(\vec{x}) \in \mathfrak{T}(\mathfrak{M})$ with variables among $\vec{x} = x_1,\ldots,x_n$ such that

$$\mathfrak{M} \models P(\underline{\vec{a}}) = \underline{\varphi(\vec{a})} \qquad \text{for all } \vec{a} \in X.$$ □

1.2.20. Observation. Let $\mathfrak{M} = (X,\cdot)$ be an applicative structure.

(i) Every representable function over $\mathfrak{M}$ is algebraic over $\mathfrak{M}$.

(ii) $\mathfrak{M}$ is combinatory complete $\Longleftrightarrow$ every algebraic function over $\mathfrak{M}$ is representable over $\mathfrak{M}$. □

The connection of combinatory completeness with combinatory logic as explained in the following proposition was implicitly present already in Schönfinkel [1924].

1.2.21. Proposition. Let $\mathfrak{M} = (X,\cdot)$ be an applicative structure. $\mathfrak{M}$ is combinatory complete $\Longleftrightarrow$ there exist elements $s,k \in X$ such that

$$\mathfrak{M} \models \underline{s}xyz = xz(yz) \wedge \underline{k}xy = x.$$

Proof: $\Rightarrow$: $(a,b,c) \mapsto ac(bc)$ and $(a,b) \mapsto a$ are just two examples of algebraic functions that should be representable, if $\mathfrak{M}$ is combinatory complete.

$\Leftarrow$: We can expand $\mathfrak{M}$ to a model of combinatory logic. Then if $P(\vec{x}) \in \mathfrak{T}(\mathfrak{M})$ let $f = [\![\langle\vec{x}\rangle P]\!]$ and use lemma 1.2.6.(ii) to show that $\mathfrak{M} \models \underline{f}\vec{x} = P$. □

1.2.22. Definition. A *combinatory algebra*, or a *model of CL*, is a structure $\mathfrak{M} = (X,\cdot,s,k)$ with $\cdot : X^2 \to X$ and $s,k \in X$ such that for all $a,b,c \in X$ we have

$$\begin{aligned} sabc &= ac(bc), \\ kab &= a. \end{aligned}$$

□

If one extends the definition of satisfaction in 1.2.17. with the obvious stipulations that $[\![S]\!]_\rho = s$ and $[\![K]\!]_\rho = k$ then $\mathfrak{M} = (X,\cdot,s,k)$ is a combinatory algebra if and only if $\mathfrak{M} \models CL$.

Much of the work in this section has been done to allow us to proceed now directly to the following important definition.

1.2.23. Definition.

(i) A *combinatory lambda algebra*, or a *model of* CL_β, is a combinatory algebra $\mathfrak{M} = (X,\cdot,s,k)$ such that

$$\mathfrak{M} \models CL_\beta.$$

(In fact, by proposition 1.2.15. this comes down to a finite extra set of axioms.)

(ii) A *combinatory lambda model*, or a *weakly extensional model of* CL_β, is a combinatory lambda algebra $\mathfrak{M}$ such that

$$\mathfrak{M} \models \xi_{CL},$$

where ξ_{CL} is the following axiomscheme

$$\forall x(P = Q) \to \langle x\rangle P = \langle x\rangle Q ,$$

with $P,Q \in \mathcal{Cl}(\mathfrak{M},x)$.

□

Just as in proposition 1.1.20. it is possible to simplify the definition of a combinatory lambda model.

1.2.24. Proposition. Let $\mathfrak{M} = (X,\cdot,s,k)$ be a combinatory algebra. Assume furthermore that

(a) $s = [\![\langle xyz\rangle .xz(yz)]\!]$,

(b) $k = [\![<xy>.x]\!]$,

(c) $\mathfrak{M} \models \xi_{CL}$.

Then $\mathfrak{M}$ is a combinatory lambda model.

Proof: We have to show that $\mathfrak{M} \models CL_\beta$.
In order to do this it is clearly sufficient to show:

(1) $\lambda \vdash M = N \Rightarrow \mathfrak{M} \models M_{CL} = N_{CL}$ for all $M,N \in \Lambda$.

(2) $\mathfrak{M} \models (P_\lambda)_{CL} = P$ for all $P \in C\ell$.

Proof of (1): By induction on the length of proof of $M = N$, using (c). To verify the β-axiom we need to know that

$$\mathfrak{M} \models M_{CL}[x := N_{CL}] = (M[x := N])_{CL}$$

for all $M,N \in \Lambda$.
This can be easily proved, again using (c), by induction on M.

Proof of (2): By induction on the structure of P, using (a) respectively (b) for the case $P \equiv S$ respectively $P \equiv K$.

□

Some more simplifications concerning ξ_{CL} will be presented in section 1.4.

We end this section by considering the term models. They form an important class of structures inbetween syntax and semantics. Although the reader readily supplies any details for himself we will define some notions for ease and future reference.

1.2.25. Definition (open term models).

Let T be a CL(C)-theory.

Let $=_T$ be the equivalence relation on $C\ell(C)$, defined by

$$P =_T Q \iff T \vdash P = Q.$$

$[P]_T$ denotes the equivalence class of P.

Let $X = C\ell(C)/=_T$ and define $\cdot : X^2 \to X$ by

$$[P]_T.[Q]_T = [PQ]_T.$$

(This does not depend on the choice of representative.)
The *open term model of T* is the structure

$$\mathfrak{M}(T) = (X,\cdot,[S]_T,[K]_T).$$

□

1.2.26. Definition (interiors).
Let $\mathfrak{M} = (X,\cdot,s,k)$ be a combinatory algebra.
Define $X^0 = \{[\![P]\!] \mid P \in \mathcal{Cl}^0\}$.
Then the *interior of* $\mathfrak{M}$ is the structure

$$\mathfrak{M}^0 = (X^0,\cdot,s,k).$$

□

Note that since $s = [\![S]\!]$, $k = [\![K]\!]$ with $S,K \in \mathcal{Cl}^0$, we have $s,k \in X^0$. Clearly also $\cdot : (X^0)^2 \to X^0$. Therefore the interior of a combinatory algebra is welldefined.

1.2.27. Lemma.

(i) If $\mathfrak{M}$ is a combinatory algebra, respectively a combinatory lambda algebra, then so is $\mathfrak{M}^0$.

(ii) There exists a combinatory lambda model $\mathfrak{M}$ such that $\mathfrak{M}^0$ is not a combinatory lambda model.

Proof:

(i) These assertions follow from the fact that CL and CL_β are equational theories.

(ii) Take $\mathfrak{M} = \mathfrak{M}(CL_\beta)$ and use the combinatory versions of the Plotkin terms of proposition 1.1.11.

□

1.2.28. Definition (closed term models).
Let T be a CL(C)-theory.
The *closed term model* of T is $\mathfrak{M}^0(T)$, the interior of the open term model of T.

□

1.2.29. Examples.

(i) $\mathfrak{M}(CL)$ and $\mathfrak{M}^0(CL)$ are both combinatory algebras, but not combinatory lambda algebras.

(ii) $\mathfrak{M}^0(CL_\beta)$ is a combinatory lambda algebra, but not a combinatory lambda model.

(iii) $\mathfrak{M}(CL_\beta)$ is a combinatory lambda model. □

1.3. *The equivalence of the crude and combinatory versions.*

The purpose of this section is to show that the approaches of sections 1.1. and 1.2., viz. the crude and the combinatory approach, are equivalent. This will be the first indication that the notion of lambda algebra (and of lambda model) is rather natural. This will be done by giving a bijective correspondence between proto lambda algebras and combinatory lambda algebras.

1.3.1. Construction. Let $(X,\cdot)$ be an applicative structure.

(i) Let $\mathfrak{M} = (X,\cdot,[\![\cdot]\!])$ be a proto lambda algebra.
Define $\mathfrak{M}' = (X,\cdot,s,k)$, where

$$s = [\![S_\lambda]\!],$$
$$k = [\![K_\lambda]\!].$$

(ii) Let $\mathfrak{N} = (X,\cdot,s,k)$ be a combinatory lambda algebra.
Define $\mathfrak{N}^+ = (X,\cdot,[\![\cdot]\!])$, where

$$[\![M]\!] = [\![M_{CL}]\!]^{\mathfrak{N}}.$$ □

Note that this construction does not change the underlying applicative structure.

The constructions ' and + are inverses of each other.

1.3.2. Theorem.

(i) Let $\mathfrak{M}$ be a proto lambda algebra.
Then $\mathfrak{M}'$ is a combinatory lambda algebra and $\mathfrak{M}'^+ = \mathfrak{M}$.

(ii) Let $\mathfrak{N}$ be a combinatory lambda algebra.
Then $\mathfrak{N}^+$ is a proto lambda algebra and $\mathfrak{N}^{+'} = \mathfrak{N}$.

Proof:

(i) Claim: For every $P \in C\ell^0(\underline{X})$ $[\![P]\!]^{\mathfrak{M}'} = [\![P_\lambda]\!]^{\mathfrak{M}}$.
Proof: This is clear by induction on the structure of P, using the definition of s and k.

Now assume $CL_\beta(\underline{X}) \vdash P = Q$, $P,Q \in \mathcal{C\ell}^0(\underline{X})$.
Then by 1.2.13.(i) $\lambda(\underline{X}) \vdash P_\lambda = Q_\lambda$.
Hence $[\![P]\!]^{\mathfrak{M}'} = [\![P_\lambda]\!]^{\mathfrak{M}} = [\![Q_\lambda]\!]^{\mathfrak{M}} = [\![Q]\!]^{\mathfrak{M}'}$.
We conclude that $\mathfrak{M}'$ is a combinatory lambda algebra.

$$\begin{aligned} \text{Furthermore } [\![M]\!]^{\mathfrak{M}'^+} &= [\![M_{CL}]\!]^{\mathfrak{M}'}, \text{ by definition,} \\ &= [\![M_{CL,\lambda}]\!]^{\mathfrak{M}}, \text{ by the claim,} \\ &= [\![M]\!]^{\mathfrak{M}}. \end{aligned}$$

Therefore $\mathfrak{M}'^+ = \mathfrak{M}$.

(ii) We use lemma 1.1.9. to show that $\mathfrak{N}^+$ is a proto lambda algebra:

Condition (a): $[\![\underline{a}]\!]^{\mathfrak{N}^+} = [\![\underline{a}_{CL}]\!]^{\mathfrak{N}} = [\![\underline{a}]\!]^{\mathfrak{N}} = a$.

$$\begin{aligned} \text{Condition (b): } [\![MN]\!] &= [\![(MN)_{CL}]\!]^{\mathfrak{N}} \\ &= [\![M_{CL}\ N_{CL}]\!]^{\mathfrak{N}} \\ &= [\![M_{CL}]\!]^{\mathfrak{N}} \cdot [\![N_{CL}]\!]^{\mathfrak{N}} \\ &= [\![M]\!] \cdot [\![N]\!]. \end{aligned}$$

Condition (c): Let $M,N \in \Lambda^0(\underline{X})$.

$$\begin{aligned} \lambda(\underline{X}) \vdash M = N \ &\Rightarrow \ CL_\beta(\underline{X}) \vdash M_{CL} = N_{CL} \\ &\Rightarrow \ [\![M_{CL}]\!]^{\mathfrak{N}} = [\![N_{CL}]\!]^{\mathfrak{N}} \\ &\Rightarrow \ [\![M]\!] = [\![N]\!]. \end{aligned}$$

Furthermore $[\![S_\lambda]\!]^{\mathfrak{N}^+} = [\![S_{\lambda,CL}]\!]^{\mathfrak{N}} = [\![S]\!]^{\mathfrak{N}} = s$ and similarly for K.
This means $\mathfrak{N}^{+\prime} = \mathfrak{N}$.

□

What is important is that not only this correspondence exists but also that it preserves the central properties of lambda algebras and models that were discussed in the first two sections, notably the notions of weak extensionality, term models and interiors.
This will be shown now.

1.3.3. Theorem. Let $\mathfrak{M}$, respectively $\mathfrak{N}$, be a proto lambda algebra, respectively combinatory lambda algebra.

(i) $\mathfrak{M} \models \xi \iff \mathfrak{M}' \models \xi_{CL}$.

(ii) $\mathfrak{N} \models \xi_{CL} \iff \mathfrak{N}^+ \models \xi$.

(iii) $(\mathfrak{M}^0)' = (\mathfrak{M}')^0$.

(iv) $(\mathfrak{N}^0)^+ = (\mathfrak{N}^+)^0$.

Proof:

(i) $\Rightarrow$: Assume $P,Q \in \mathcal{C\ell}(\underline{X})$, $FV(PQ) \subseteq \{x\}$ and $\mathfrak{M}' \models \forall x(P = Q)$.
Then $\mathfrak{M} \models \forall x(P_\lambda = Q_\lambda)$. Since $\mathfrak{M} \models \xi$, we have
$\mathfrak{M} \models \lambda x.P_\lambda = \lambda x.Q_\lambda$. Equivalently, by 1.2.9.(i),
$\mathfrak{M} \models (\langle x\rangle P)_\lambda = (\langle x\rangle .Q)_\lambda$.
Therefore $\mathfrak{M}' \models \langle x\rangle P = \langle x\rangle Q$.

(ii) $\Rightarrow$: Similarly.

(i)(ii) $\Leftarrow$: These follow from (i)(ii) $\Rightarrow$ by using $\mathfrak{M}'^+ = \mathfrak{M}$ and $\mathfrak{N}^{+\prime} = \mathfrak{N}$.

(iii) Let $\mathfrak{M} = (X,\cdot,[\![\cdot]\!])$.
Then $\mathfrak{M}^0 = (X_1^0,\cdot,[\![\cdot]\!])$, $X_1^0 = \{[\![M]\!] \mid M \in \Lambda^0\}$.
$(\mathfrak{M}^0)' = (X_1^0,\cdot,[\![S_\lambda]\!],[\![K_\lambda]\!])$.
Furthermore $\mathfrak{M}' = (X,\cdot,[\![S_\lambda]\!],[\![K_\lambda]\!])$,
$(\mathfrak{M}')^0 = (X_2^0,\cdot,[\![S_\lambda]\!],[\![K_\lambda]\!])$, $X_2^0 = \{[\![P]\!]^{\mathfrak{M}'} \mid P \in \mathcal{C\ell}^0\}$.

It remains to be shown, that $X_1^0 = X_2^0$.
But $[\![P]\!]^{\mathfrak{M}'} = [\![P_\lambda]\!]$ and $[\![M]\!] = [\![M_{CL,\lambda}]\!]$, hence this is clear.

(iv) We have $(\mathfrak{N}^0)^+ = (((\mathfrak{N}^+)')^0)^+$, by theorem 1.3.2.,
$= (((\mathfrak{N}^+)^0)')^+$, by (iii),
$= (\mathfrak{N}^+)^0$, by theorem 1.3.2.

□

Also term models correspond; to show this we need the obvious correspondence between theories.

1.3.4. Definition.

(i) If T is a λ-theory, let T_{CL} be the CL_β-theory, defined by
$T_{CL} = \{P = Q \mid (P_\lambda = Q_\lambda) \in T\}$.

(ii) If T is a CL_β-theory, let T_λ be the λ-theory defined by
$T_\lambda = \{M = N \mid (M_{CL} = N_{CL}) \in T\}$.

□

By looking more carefully to the equivalence of combinatory logic and lambda calculus as studied in section 1.2. it can be shown that T_{CL} is really a CL_β-theory, T_λ really a λ-theory and that $T_{CL,\lambda} = T$, $T_{\lambda,CL} = T$.

1.3.5. Proposition.

(i) If T is a λ-theory, then $(\mathfrak{M}(T))' \cong \mathfrak{M}(T_{CL})$.

(ii) If T is a CL-theory, then $(\mathfrak{M}(T))^+ \cong \mathfrak{M}(T_\lambda)$.

(The notion of isomorphism, $\cong$, is the obvious one. See also section 1.6.)

Proof:

(i) There is the obvious correspondence $[M]_T \leftrightarrow [M_{CL}]_{T_{CL}}$.

(ii) Use the correspondence $[P]_T \leftrightarrow [P_\lambda]_{T_\lambda}$.

□

Now that we have proven the equivalence of the combinatory and crude approaches, we may state the following convention.

1.3.6. Convention. From now on, when we talk about lambda algebras or lambda models, we move freely between the combinatory and crude versions of the theory, if no confusion arises. There will be no essential difference between M and M_{CL}, or P and P_λ. Depending on the context it should be clear what is meant. Both combinatory lambda algebras and proto lambda algebras will be referred to as just lambda algebras. The same convention applies to lambda models. Moreover the following abuses of notation will be used frequently.

Let $\mathfrak{M} = (X, \cdot, [\![\,.\,]\!])$ be a lambda algebra.

$\mathfrak{M}$ may stand for $\mathfrak{M}$, $|\mathfrak{M}|$ or X.

M may stand for M, $[\![M]\!]$ or $[\![M_{CL}]\!]$.

a may stand for a or $\underline{a}$.

For example $\lambda x.ax \in \mathfrak{M}$ stands for

$$[\![\lambda x.\underline{ax}]\!] \in |\mathfrak{M}|.$$

□

Let us state a few properties of the by now familiar combinatory algebras, lambda algebras and lambda models.

1.3.7. Proposition.

(i) {combinatory algebras} $\subseteq$ {lambda algebras} $\subseteq$ {lambda models}.

(ii) $(|\mathfrak{M}(CL)|, \cdot)$ is an applicative structure, that can be expanded to a combinatory algebra, but not to a lambda algebra.

(iii) $(|\mathfrak{M}^0(\mathcal{H})|, \cdot)$ is an applicative structure, that can be expanded to a lambda algebra, viz. $\mathfrak{M}^0(\mathcal{H})$, but not to a lambda model. ($\mathcal{H}$ is the λ-theory, equating all unsolvables, see Barendregt [1981].)

Proof: See Barendregt and Koymans [1980].

□

For more relations between lambda algebras and lambda models, see section 1.6.

1.4. *Lambda models.*

The structures that are most easy to handle are certainly the lambda models. The first real mathematical lambda calculus structures that were discovered, viz. $\mathbb{D}_\infty$ of Scott [1972] and $\mathbb{P}\omega$ of Plotkin [1972] and Scott [1974], were both weakly extensional and hence lambda models. $\mathbb{D}_\infty$ was even extensional, see section 1.5. The most important advantage of lambda models is that lambda terms can be interpreted as real functions; these are extensionally characterized by their applicative behaviour in contrast with the notion of algorithm that has an intensional character. We approached lambda models in sections 1.1. and 1.2. via the detour of lambda algebras. The purpose of this section is to simplify the description of lambda models and to concentrate on their particular properties.

The main idea for carrying out this program, is making full use of the strength of the axiom of weak extensionality. First of all we want to restate this axiom in such a way that it is first-order, with only application as a nonlogical symbol, besides the constants S and K.

1.4.1. Definition.

(i) $1_n = \lambda xy_1\ldots y_n.xy_1\ldots y_n$.

(ii) $I = 1_0$, $1 = 1_1$.

Let $\mathfrak{M}$ be a combinatory algebra.

(iii) We use 1_n also for $[\![<xy_1\ldots y_n> xy_1\ldots y_n]\!]^{\mathfrak{M}}$,
$i = [\![I]\!]$, $1 = [\![1]\!]$.

(iv) The Meyer-Scott axiom is the following statement

$$\forall xy(\forall z(xz = yz) \to 1x = 1y) \qquad (\mathrm{MS}).$$

□

1.4.2. Lemma. Let $\mathfrak{M}$ be a combinatory algebra.

(i) $\mathfrak{M} \models \xi_{CL} \Rightarrow \mathfrak{M} \models \mathrm{MS}$.

(ii) If $\mathfrak{M} \models 1(\langle x\rangle P) = \langle x\rangle P$ for all $P \in \mathcal{C}\ell^0(\mathfrak{M},x)$ then

$$\mathfrak{M} \models \mathrm{MS} \Rightarrow \mathfrak{M} \models \xi_{CL}.$$

(iii) If $\mathfrak{M}$ is a lambda algebra then

$$\mathfrak{M} \models \xi_{CL} \iff \mathfrak{M} \models \mathrm{MS}.$$

Proof:

(i) Suppose $\mathfrak{M} \models \xi_{CL}$, $a,b \in |\mathfrak{M}|$ and $\mathfrak{M} \models \forall z(\underline{a}z = \underline{b}z)$.
By (ξ_{CL}) $\mathfrak{M} \models \langle z\rangle \underline{a}z = \langle z\rangle \underline{b}z$.
Hence $\mathfrak{M} \models 1\underline{a} = 1\underline{b}$.
We conclude that $\mathfrak{M} \models \mathrm{MS}$.

(ii) Suppose $\mathfrak{M} \models \mathrm{MS}$ and $\mathfrak{M} \models \forall x(P = Q)$ with $P,Q \in \mathcal{C}\ell^0(\mathfrak{M},x)$. Let $a = [\![\langle x\rangle.P]\!]$, $b = [\![\langle x\rangle.Q]\!]$. Then $\mathfrak{M} \models \forall z(\underline{a}z = \underline{b}z)$.
By (MS) $\mathfrak{M} \models 1\underline{a} = 1\underline{b}$, that is $\mathfrak{M} \models 1(\langle x\rangle P) = 1(\langle x\rangle Q)$.
By the assumption about $\mathfrak{M}$ this implies

$$\mathfrak{M} \models \langle x\rangle P = \langle x\rangle Q .$$

Therefore $\mathfrak{M} \models \xi_{CL}$ has been shown.

(iii) Suppose $\mathfrak{M}$ is a lambda algebra.
Since $\mathrm{CL}_\beta(\mathfrak{M}) \vdash 1(\langle x\rangle P) = \langle x\rangle P$ for all $P \in \mathcal{C}\ell^0(\mathfrak{M},x)$ it follows that $\mathfrak{M} \models 1(\langle x\rangle P) = \langle x\rangle P$ for all $P \in \mathcal{C}\ell^0(\mathfrak{M},x)$.
The assertion now follows from (i) and (ii).

□

The following observation will show us a natural way to produce a theorem of Meyer [1980]. We would like to replace ξ_{CL} by MS. But in the assumptions of lemma 1.4.2.(iii) we still need that $\mathfrak{M}$ is a lambda algebra and hence that $\mathfrak{M}$ satisfies the rather complex axiomset A_β. This axiomset can be eliminated in the style of proposition 1.2.24., replacing ξ_{CL} by MS. But does the conclusion of lemma 1.4.2.(iii) still hold then? Looking at lemma 1.4.2.(ii) we see that this holds for an arbitrary combinatory algebra, if only we are able to prove that $\mathfrak{M} \models 1(\langle x\rangle P) = \langle x\rangle P$ for all $P \in \mathcal{C}\ell^0(\mathfrak{M},x)$. But this can easily be attained by the use of two extra axioms.

1.4.3. Theorem (adaptation of Meyer [1980]).

Let $\mathfrak{M} = (X, \cdot, s, k)$ be a combinatory algebra such that

(a) $s = 1_3 s$,

(b) $k = 1_2 k$,

(c) $\mathfrak{M} \models MS$.

Then $\mathfrak{M}$ is a lambda model.

Proof: For any $Q, R \in C\ell^0(\underline{X})$ we have

$\mathfrak{M} \models SQR = 1_3 SQR = \langle x \rangle . SQRx = 1(SQR)$ and

$\mathfrak{M} \models KQ \ = 1_2 KQ \ = \langle x \rangle . KQx \ = 1(KQ)$.

Since $(\langle x \rangle P)$ is always of the form SQR or KQ, we may conclude

$\mathfrak{M} \models 1(\langle x \rangle P) = \langle x \rangle P$, for all $P \in C\ell^0(\underline{X}, x)$.

By (c) and lemma 1.4.2.(ii) we have $\mathfrak{M} \models \xi_{CL}$.

Now we can finish the proof by applying proposition 1.2.24. For by the now established validity of ξ_{CL} we have

$$s = 1_3 s = \langle xyz \rangle . sxyz = \langle xyz \rangle . xz(yz),$$
$$k = 1_2 k = \langle xy \rangle . kxy = \langle xy \rangle . x.$$

□

In the original version of this theorem, Meyer used other expressions than 1_2 and 1_3 which have the advantage of being shorter (when written out in s and k), but the disadvantage that the proofs involved get more complicated. Scott [1980] used even shorter expressions. An important difference is that Meyer's method tries to avoid any specific properties of s and k, in contrast to Scott's method, cf. corollary 1.4.10. Now we will compare these two methods.

1.4.4. Definition. Let $\mathfrak{M} = (X, \cdot, s, k)$ be a combinatory algebra and $\varepsilon \in X$.

(i) (Meyer) $(\varepsilon_n^M)_{n \in \omega}$ is inductively defined by

$$\varepsilon_1^M = \varepsilon, \ \varepsilon_{n+1}^M = s(k\varepsilon)(s(k\varepsilon_n^M)).$$

(ii) (Scott) $(\varepsilon_n^S)_{n \in \omega}$ is inductively defined by

$$\varepsilon_1^S = \varepsilon, \ \varepsilon_{n+1}^S = s(k\varepsilon_n^S).$$

(iii) MS_ε is the Meyer-Scott axiom where 1 is replaced by ε, that is

$$\forall xy(\forall z(xz = yz) \rightarrow \varepsilon x = \varepsilon y).$$

□

1.4.5.A. Lemma (Meyer's case).

Let $\mathfrak{M} = (X,\cdot,s,k)$ be a combinatory algebra and $\varepsilon \in X$.

If $\mathfrak{M} \models MS_\varepsilon$ and $\forall ab \in X(\varepsilon ab = ab)$, then for all $a,b \in X$, $n \geqslant 1$ (writing ε_n for ε_n^M)

(a) $\varepsilon_{n+1}ab = \varepsilon_n(ab)$,

(b) $\varepsilon(\varepsilon_n a) = \varepsilon_n a$,

(c) $\forall x_1 \ldots x_n \in X(ax_1 \ldots x_n = bx_1 \ldots x_n) \leftrightarrow \varepsilon_n a = \varepsilon_n b$.

Proof:

(a) $$\begin{aligned}\varepsilon_{n+1}ab &= s(k\varepsilon)(s(k\varepsilon_n))ab\\ &= \varepsilon(s(k\varepsilon_n)a)b\\ &= s(k\varepsilon_n)ab\\ &= \varepsilon_n(ab).\end{aligned}$$

(b) If $n = 1$: $\varepsilon ax = ax$ for all x,

hence $\varepsilon(\varepsilon a) = \varepsilon a$.

If $n > 1$: $$\begin{aligned}\varepsilon_n a &= s(k\varepsilon)(s(k\varepsilon_{n-1}))a\\ &= \varepsilon(s(k\varepsilon_{n-1})a)\end{aligned}$$

and hence the result follows by the case $n = 1$.

(c) $\leftarrow$: By induction on $n \geqslant 1$ it follows from (a) that

$$\varepsilon_n ax_1 \ldots x_n = ax_1 \ldots x_n.$$

Hence the result.

$\rightarrow$: By induction on $n \geqslant 1$:

Basis: For $n = 1$ it holds by assumption.

Induction step: Assume $ax_1x_2 \ldots x_{n+1} = bx_1x_2 \ldots x_{n+1}$.

Then by IH, $\varepsilon_n(ax_1) = \varepsilon_n(bx_1)$.

By (a) we have $\varepsilon_{n+1}ax_1 = \varepsilon_{n+1}bx_1$.

Therefore $\varepsilon(\varepsilon_{n+1}a) = \varepsilon(\varepsilon_{n+1}b)$.

By (b) finally $\varepsilon_{n+1}a = \varepsilon_{n+1}b$.

□

In Scott's case the proof of the corresponding lemma is more complicated and uses a special property of s. On the other hand it does not need the assumption that $\varepsilon ab = ab$ for all $a,b \in X$.

1.4.5.B. Lemma (Scott's case).

Let $\mathfrak{M} = (X,\cdot,s,k)$ be a combinatory algebra and $\varepsilon \in X$.

If $\mathfrak{M} \models MS_{\varepsilon}$ and $s = \varepsilon_3 s$, then for all $a,b \in X$, $n \geqslant 1$ (writing ε_n for ε_n^S)

(a) $\varepsilon_{n+1}ab = \varepsilon_n(ab)$,

(b1) $\varepsilon(sab) = sab$,

(b2) $\varepsilon a = 1a$,

(b3) $\varepsilon(\varepsilon_n a) = \varepsilon_n a$,

(c) $\forall x_1 \ldots x_n \in X(a\vec{x} = b\vec{x}) \leftrightarrow \varepsilon_n a = \varepsilon_n b$.

Proof:

(a) $\varepsilon_{n+1}ab = s(k\varepsilon_n)ab$
$= \varepsilon_n(ab)$.

(b1) $sab = \varepsilon_3 sab$
$= \varepsilon_2(sa)b$
$= \varepsilon(sab)$.

(b2) Since for all $x \in X$ we have $1ax = ax$ we conclude by MS_{ε} that $\varepsilon(1a) = \varepsilon a$.
On the other hand, since $1a = \langle x \rangle ax = s(ka)i$ we may apply (b1) to show $\varepsilon(1a) = 1a$.
Hence $\varepsilon a = 1a$.

(b3) For $n = 1$: $\varepsilon ax = 1ax = ax$, by (b2).
Therefore $\varepsilon(\varepsilon a) = \varepsilon a$ by MS_{ε}.
For $n > 1$: $\varepsilon_n a = s(k\varepsilon_{n-1})a$
$= \varepsilon(s(k\varepsilon_{n-1})a)$, by (b1),
$= \varepsilon(\varepsilon_n a)$.

(c) Follows from (a) and (b3) exactly as in lemma 1.4.5.A.

□

Now we will proceed with a quite general lemma that is applicable to both Meyer's and Scott's case.

1.4.6. Lemma. Let $\mathfrak{M} = (X,\cdot,s,k)$ be a combinatory algebra and $(\varepsilon_n)_{n \geqslant 1}$ an arbitrary sequence of elements of X. If the following conditions are satisfied for all $a,b \in X$, $n \geqslant 1$

(a) $s = \varepsilon_3 s$,

(b) $k = \varepsilon_2 k$,

(c) $\mathfrak{M} \models MS_{\varepsilon_1}$,

(d) $\varepsilon_{n+1}ab = \varepsilon_n(ab)$,
(e) $\varepsilon_1(\varepsilon_n a) = \varepsilon_n a$,

then $\mathfrak{M}$ is a lambda model and $1_n = \varepsilon_1\varepsilon_n = 1\varepsilon_n$ for all $n \geqslant 1$.

Proof: We will show that $\forall n \geqslant 1\ (\varepsilon_n a = 1_n a)$ (1).
Then the first part of the theorem easily follows because (a), (b) and (c) transform into $s = 1_3 s$, $k = 1_2 k$ and $\mathfrak{M} \models MS$ and therefore theorem 1.4.3. applies.
In order to prove (1) we first show

$\varepsilon_1(1_n a) = 1_n a$ (2).

Proof of (2): We have $sab = \varepsilon_3 sab$
$= \varepsilon_2(sa)b$
$= \varepsilon_1(sab)$
and similarly $ka = \varepsilon_1(ka)$.
Therefore, since any $\langle x\rangle P$ is of the form SQR or KQ we conclude $\varepsilon_1(\langle x\rangle P) = \langle x\rangle P$ for all $P \in \mathcal{C}\ell^0(\underline{X},x)$.
In particular $\varepsilon_1(1_n a) = 1_n a$.
End of proof of (2).

Proof of (1): We proceed by induction on n.
Basis: $ax = 1ax$ for all $x \in X$, hence
$\varepsilon_1 a = \varepsilon_1(1a)$ by MS_{ε_1}, hence
$\varepsilon_1 a = 1a$ by (2).
Induction step: $\varepsilon_{n+1}ax = \varepsilon_n(ax)$, by (d),
$= 1_n(ax)$, by IH,
$= 1_{n+1}ax$, for all $x \in X$.
Hence $\varepsilon_1(\varepsilon_{n+1}a) = \varepsilon_1(1_{n+1}a)$ by MS_{ε_1}.
This means $\varepsilon_{n+1}a = 1_{n+1}a$ by (e),(2).
End of proof of (1).

The second assertion in the theorem, viz. $1_n = \varepsilon_1\varepsilon_n = 1\varepsilon_n$, follows easily: Apply MS to (1) to get $1\varepsilon_n = 11_n$. But $1\varepsilon_n = \varepsilon_1\varepsilon_n$ by (1) and $11_n = 1_n$ because $\mathfrak{M}$ is a lambda model.

□

Now we can put these results together.

1.4.7. Theorem (Meyer [1980], Scott [1980]).

Let $\mathfrak{M} = (X,\cdot,s,k)$ be a combinatory algebra and $\varepsilon \in X$.

Let $(\varepsilon_n)_{n\geqslant 1}$ be either $(\varepsilon_n^M)_{n\geqslant 1}$ or $(\varepsilon_n^S)_{n\geqslant 1}$.

If (a) $s = \varepsilon_3 s$,

(b) $k = \varepsilon_2 k$,

(c) $\mathfrak{M} \models MS_\varepsilon$,

(d) (Only in Meyer's case) $\varepsilon ab = ab$ for all $a,b \in X$,

then $\mathfrak{M}$ is a lambda model and $1_n = \varepsilon\varepsilon_n = 1\varepsilon_n$ for all $n \geqslant 1$.

Proof: Apply lemma 1.4.6. together with lemma 1.4.5.A. (for Meyer's case) or lemma 1.4.5.B. (for Scott's case).

□

Although Meyer's definition of $(\varepsilon_n)_{n\geqslant 1}$, is a little more complicated than Scott's definition, it has the advantage that with it we may avoid any extra properties of s,k (as e.g. $s = \varepsilon_3 s$) in the definition of a lambda model. This will be shown in 1.4.8. - 1.4.10. From now on ε_n means ε_n^M.

1.4.8. Definition.

(i) A *combinatory algebra with* ε is a structure $\mathfrak{M} = (X,\cdot,s,k,\varepsilon)$ such that $(X,\cdot,s,k)$ is a combinatory algebra and ε is such that $\mathfrak{M} \models MS_\varepsilon$ and $\forall ab \in X(\varepsilon ab = ab)$.

(ii) Let $\mathfrak{M}$ be a combinatory algebra with ε.
Define $\Phi(\mathfrak{M}) = (X,\cdot,\bar{s},\bar{k})$, where $\bar{s} = \varepsilon_3 s$ and $\bar{k} = \varepsilon_2 k$.

(iii) Let $\mathfrak{N} = (X,\cdot,s,k)$ be a lambda model.
Define $\Psi(\mathfrak{N}) = (X,\cdot,s,k,1)$.

□

1.4.9. Proposition. Let $\mathfrak{M} = (X,\cdot,s,k,\varepsilon)$ and $\mathfrak{M}' = (X,\cdot,s',k',\varepsilon')$ be two combinatory algebras with ε (with the same underlying applicative structure).

(i) $\Phi(\mathfrak{M}) = \Phi(\mathfrak{M}') \iff \varepsilon\varepsilon = \varepsilon'\varepsilon'$.

(ii) $\Phi(\mathfrak{M})$ is a lambda model with $\bar{1} =_{\mathrm{def}} [\![1]\!]^{\Phi(\mathfrak{M})} = \varepsilon\varepsilon$.

Proof: It is easy to see that $\Phi(\mathfrak{M})$ and $\Phi(\mathfrak{M}')$ are combinatory algebras.

(i) $\Leftarrow$: Suppose $\varepsilon\varepsilon = \varepsilon'\varepsilon'$. We will show by induction on n that

$\varepsilon_n a = \varepsilon_n' a$ for all $a \in X$ (*).

Basis: $\varepsilon a = \varepsilon\varepsilon a = \varepsilon'\varepsilon' a = \varepsilon' a$ (**).

Induction step: $\varepsilon_{n+1}ax = \varepsilon_n(ax)$, by 1.4.5.A.(a),
$= \varepsilon'_n(ax)$, by IH,
$= \varepsilon'_{n+1}ax$, by 1.4.5.A.(a).
Therefore by MS_ε, $\varepsilon(\varepsilon_{n+1}a) = \varepsilon(\varepsilon'_{n+1}a)$.
By (**), $\varepsilon(\varepsilon'_{n+1}a) = \varepsilon'(\varepsilon'_{n+1}a)$.
Therefore $\varepsilon(\varepsilon_{n+1}a) = \varepsilon'(\varepsilon'_{n+1}a)$ and making use of lemma 1.4.5.A.(b) we may conclude $\varepsilon_{n+1}a = \varepsilon'_{n+1}a$. This proves (*).

Now $\varepsilon_3 s = \varepsilon'_3 s$, by (*),
$= \varepsilon'_3 s'$, by lemma 1.4.5.A.(c).
Similarly $\varepsilon_2 k = \varepsilon'_2 k'$.
We conclude $\Phi(\mathfrak{M}) = \Phi(\mathfrak{M}')$.

(ii) : We apply (i) $\Leftarrow$ to $\mathfrak{M}$ and $\mathfrak{M}' \equiv (\Phi(\mathfrak{M}),\varepsilon)$. Clearly, since now $\varepsilon' = \varepsilon$, we have $\varepsilon\varepsilon = \varepsilon'\varepsilon'$. Therefore (by (i) $\Leftarrow$) $\Phi(\mathfrak{M}) = \Phi(\mathfrak{M}')$. But this just tells us that the conditions for theorem 1.4.7. are fulfilled in the case of $\Phi(\mathfrak{M})$. Therefore $\Phi(\mathfrak{M})$ is a lambda model and moreover $\bar{1} = \varepsilon\varepsilon$.

(i) $\Rightarrow$: Suppose $\Phi(\mathfrak{M}) = \Phi(\mathfrak{M}')$. Then clearly by (ii)

$$\varepsilon\varepsilon = [\![1]\!]^{\Phi(\mathfrak{M})} = [\![1]\!]^{\Phi(\mathfrak{M}')} = \varepsilon'\varepsilon'.$$

□

Let us remark here that the other way around is easy. If $\mathfrak{N}$ is a lambda model, then $\Psi(\mathfrak{N})$ is a combinatory algebra with ε and $\Phi(\Psi(\mathfrak{N})) = \mathfrak{N}$. Hence (Ψ,Φ) defines the class of lambda models as a "retract" of the class of combinatory algebras with ε.

1.4.10. Corollary (Meyer [1980]).
Let $\mathfrak{M} = (X,\cdot)$ be a combinatory complete applicative structure and $\varepsilon \in X$. If

(a) $\mathfrak{M} \models MS_\varepsilon$,
(b) $\forall ab \in X\ (\varepsilon ab = ab)$,
(c) $\varepsilon\varepsilon = \varepsilon$,

then there exists a unique pair $(\bar{s},\bar{k})$ such that $(X,\cdot,\bar{s},\bar{k})$ is a lambda model with $\bar{1} = \varepsilon$.

Proof: Choose arbitrary $s,k \in X$ such that $\mathfrak{M}^* = (X,\cdot,s,k,\varepsilon)$ is a combinatory algebra with ε. Then $\Phi(\mathfrak{M}^*) = (X,\cdot,\bar{s},\bar{k})$ is a lambda model with $\bar{1} = \varepsilon\varepsilon = \varepsilon$. Suppose $\mathfrak{N} = (X,\cdot,s',k')$ is another such lambda model with $1' = \varepsilon$. Then $\mathfrak{N} = \Phi(\Psi(\mathfrak{N})) = \Phi(\mathfrak{M}^*)$ by proposition 1.4.9.(i).

□

We will refer to a structure $(X,\cdot,\varepsilon)$ satisfying 1.4.10. (a),(b),(c) as a *Meyer lambda model*. This gives one more example of the many equivalent definitions of the concept lambda model. The most important feature of this description is that it is first-order and moreover relatively simple, as compared to the equational definition of lambda algebras in section 1.2.

In order to get to the easiest definition of a lambda model we should leave the realms of combinatory manipulations and look at models as naively as possible. The only defect is that this approach is not first-order.

1.4.11. Definition. Let $(X,\cdot)$ be an applicative structure.

(i) $[X^n \to X] = \{f: X^n \to X \mid f \text{ is representable}\}$.

(ii) $F: X \to [X \to X]$ is defined by $F(a)(b) = a \cdot b$.

(iii) $(X,\cdot,F,G)$ is a *functional domain* if $G: [X \to X] \to X$ is a map such that $F \circ G = id_{[X\to X]}$.

□

1.4.12. Remarks.

(i) $\cdot$ and F are interdefinable and therefore we will also talk about functional domains $(X,\cdot,G)$ or (X,F,G).

(ii) The basic idea here is that the function G chooses a canonical representative for any function that is representable.

(iii) We call F also the function-operator or fun-operator and G the graph-operator.

□

Let us try to build an interpretation for lambda terms in a functional domain.

1.4.13. Definition. Let $\mathfrak{M} = (X,\cdot,G)$ be a functional domain.

(i) Define a partial map $[\![\cdot]\!]: \Lambda^0(\underline{X}) \rightharpoonup X$ by:

$[\![\underline{d}]\!] = d$, for $d \in X$;

$$\begin{array}{lll} \llbracket MN \rrbracket & = \llbracket M \rrbracket \cdot \llbracket N \rrbracket & \text{, if both } \llbracket M \rrbracket, \llbracket N \rrbracket \downarrow \text{ ,} \\ & = \uparrow & \text{, otherwise;} \\ \llbracket \lambda x.M \rrbracket & = G(d \mapsto \llbracket M(\underline{d}) \rrbracket) \text{ ,} & \text{if for all } d \text{ one has } \llbracket M(\underline{d}) \rrbracket \downarrow \\ & & \text{and } (d \mapsto \llbracket M(\underline{d}) \rrbracket) \in [X \to X] \text{,} \\ & = \uparrow & \text{, otherwise.} \end{array}$$

(ii) $\mathfrak{M}$ is called a (*functional*) *lambda model*, if $\llbracket . \rrbracket$ is a total map.

□

1.4.14. <u>Proposition</u>.

(i) For each functional lambda model $\mathfrak{M} = (X, \cdot, G)$ the corresponding $\mathfrak{M} = (X, \cdot, \llbracket \cdot \rrbracket)$ is a lambda model.

(ii) A lambda model $\mathfrak{M} = (X, \cdot, \llbracket . \rrbracket)$ defines a functional lambda model by putting $G(f) = \llbracket \lambda x. \underline{ax} \rrbracket$, where $a \in X$ is any representative of $f \in [X \to X]$.

(iii) The constructions in (i) and (ii) are each others inverse.

Proof:

(i) Routine to verify (β) and (ξ).

(ii) Note that the definition of G does not depend on the choice of a because of weak extensionality. $F \circ G = \mathrm{id}$ is easy. As to totality, let the partial map defined by G be $\llbracket \cdot \rrbracket^G$. Then one shows by induction on the structure of $M \in \Lambda^0(\underline{X})$ that $\llbracket M \rrbracket^G \downarrow$ and $\llbracket M \rrbracket^G = \llbracket M \rrbracket$. The only difficult case is $M = \lambda x.N$. Consider the function $d \mapsto \llbracket N(\underline{d}) \rrbracket^G = \llbracket N(\underline{d}) \rrbracket$. This function is representable by $\llbracket \lambda x.N \rrbracket$. Hence $\llbracket \lambda x.N \rrbracket^G$ is defined and we have $\llbracket \lambda x.N \rrbracket^G = \llbracket \lambda x. \underline{\llbracket \lambda x.N \rrbracket}\, x \rrbracket = \llbracket \lambda x.N \rrbracket$.

(iii) As we just saw in (ii), starting with a lambda model brings us back to the same lambda model. Now let $\mathfrak{M} = (X, \cdot, G)$ be a functional lambda model and $\mathfrak{M}' = (X, \cdot, G')$ the result of transforming $\mathfrak{M}$ into a lambda model and back to a functional lambda model again. Let $f \in [X \to X]$, representable by $a \in X$.

$$\begin{aligned} \text{Then } G'(f) &= \llbracket \lambda x. \underline{a} x \rrbracket \\ &= G(d \mapsto \llbracket \underline{a}\, \underline{d} \rrbracket) \\ &= G(d \mapsto a.d) \\ &= G(f). \end{aligned}$$

Hence $\mathfrak{M} = \mathfrak{M}'$.

□

Because of this proposition, functional lambda models will be called also just lambda models.

The following proposition gives a connection between the notions of functional lambda model and combinatory algebra with ε.

1.4.15. Proposition. Let $\mathfrak{M} = (X, \cdot, F, G)$ be a functional domain.
Then $\mathfrak{M}$ is a lambda model $\iff$

(a) $(X, \cdot)$ is combinatory complete, and

(b) $G \circ F \in [X \to X]$.

In this case we have $1 = G(G \circ F)$.

Proof:

$\Rightarrow$: Assume $\mathfrak{M}$ is a lambda model.
Then (a) is clearly satisfied. To show (b) let us compute $\llbracket 1 \rrbracket$ in $\mathfrak{M}$:
For any $d \in |\mathfrak{M}|$ we have

$$\begin{aligned} \llbracket \lambda y.\underline{d}\, y \rrbracket &= G(e \mapsto \llbracket \underline{d}\, \underline{e} \rrbracket) \\ &= G(e \mapsto de) \\ &= G(F(d)). \end{aligned}$$

$\llbracket \lambda xy.xy \rrbracket \downarrow$ means that $(d \mapsto \llbracket \lambda y.\underline{d}\, y \rrbracket) = G \circ F \in [X \to X]$.
Moreover we have $1 = G(G \circ F)$.

$\Leftarrow$: Assume (a),(b). It is easy to show by induction on $M(\vec{x}) \in \Lambda(\underline{X})$ with $FV(M) \subseteq \vec{x} = x_1 \ldots x_n$ that

(1) for all $\vec{d} \in X^n$ $\llbracket M(\vec{\underline{d}}) \rrbracket \downarrow$ and

(2) $(\vec{d} \mapsto \llbracket M(\vec{\underline{d}}) \rrbracket) \in [X^n \to X]$.

Let me do the hardest case:
Assume $M(\vec{x}) = \lambda y.N(\vec{x}, y)$. By induction hypothesis we know that $\forall \vec{d} \in X^n\ \forall e \in X\ \ \llbracket N(\vec{\underline{d}}, \underline{e}) \rrbracket \downarrow$ and moreover that there exists some $a \in X$ such that $a\vec{d}e = \llbracket N(\vec{\underline{d}}, \underline{e}) \rrbracket$ for all $\vec{d} \in X^n, e \in X$.

Then $(e \mapsto a\vec{d}e) = F(a\vec{d}) \in [X \to X]$ and hence $\llbracket M(\vec{\underline{d}}) \rrbracket = G(F(a\vec{d})) = \varepsilon(a\vec{d})$, where $\varepsilon = G(G \circ F)$. Therefore $\llbracket M(\vec{\underline{d}}) \rrbracket \downarrow$ for all $\vec{d} \in X^n$ and this shows (1). Moreover $\vec{d} \mapsto \varepsilon(a\vec{d})$ is algebraic and hence representable by (a). This shows (2).

□

1.4.16. Corollary. There exists a bijective correspondence between functional lambda models $(X,\cdot,G)$ and Meyer lambda models $(X,\cdot,\varepsilon)$, given by $\varepsilon = G(G\circ F)$ and $G(F(a)) = \varepsilon a$.

Proof: Combine corollary 1.4.10. and propositions 1.4.14., 15. A direct proof can also be given.

□

Finally we want to talk about a fifth equivalent way of defining a lambda model, originating with Scott, and intimately connected with the functional domain approach. As was noticed by Scott the graph-operator G in a functional domain is completely determined by its range, being a subset of X.

1.4.17. Lemma. Let $(X,\cdot,G)$ and $(X,\cdot,G')$ be functional domains. If $\mathrm{Ran}(G) = \mathrm{Ran}(G')$ then $G = G'$.

Proof: Suppose $\mathrm{Ran}(G) = \mathrm{Ran}(G')$ and let $f \in [X\to X]$. We want to show $G(f) = G'(f)$. Since $\mathrm{Ran}(G) = \mathrm{Ran}(G')$ we can pick $g \in [X\to X]$ such that $G(f) = G'(g)$. It suffices to show that $f = g$. Therefore pick $a \in X$. Then $f(a) = G(f).a = G'(g).a = g(a)$.

□

Scott's axiomatization of lambda models can be given now by translating the properties of G in proposition 1.4.15. into properties of $R = \mathrm{Ran}(G)$.

1.4.18. Definition. Let $(X,\cdot)$ be a combinatory complete applicative structure and $R \subseteq X$. $(X,\cdot,R)$ is a *Scott lambda model* if

(a) $\forall a,b \in R\ (\forall x \in X(ax = bx) \to a = b)$

and there exists an element $\varepsilon \in X$ such that

(b) $\forall x \in X\ (\varepsilon x \in R)$,

(c) $\forall x,y \in X\ (\varepsilon xy = xy)$.

□

1.4.19. Proposition. There exists a bijective correspondence between functional lambda models $(X,\cdot,G)$ and Scott lambda models $(X,\cdot,R)$ given by $R = \mathrm{Ran}(G)$ and

$G(f) = \varepsilon a$, where ε is any element of X, satisfying 1.4.18.(b),(c) and a is any element of X, representing f.

Proof: Routine. □

Scott's original definition of a Scott lambda model made use of the combinators s and k instead of the combinator 1.

1.4.20. Definition. Let $(X,\cdot)$ be an applicative structure and $R\subseteq X$. Define $R_1 = R$ and $R_{n+1} = \{a\in R \mid \forall x\in X(ax\in R_n)\}$, $n\geq 1$. □

1.4.21. Proposition. Let $(X,\cdot)$ be an applicative structure and $R\subseteq X$. Then $(X,\cdot,R)$ is a Scott lambda model $\iff$

(a) $\forall a,b\in R(\forall x\in X(ax=bx)\to a=b)$ and

there exist elements $s\in R_3$, $k\in R_2$ such that

(b) $sxyz = xz(yz)$, for all $x,y,z\in X$,

(c) $kxy = x$, for all $x,y\in X$.

Proof:

$\Rightarrow$: Suppose $(X,\cdot,R)$ is a Scott lambda model, corresponding to the ordinary lambda model $(X,\cdot,s,k)$. It is easy to check that $s\in R_3$, $k\in R_2$ because in general

$$R_n = \{a\in X \mid 1_n a = a\}.$$

$\Leftarrow$: Let $s\in R_3$, $k\in R_2$ satisfy (b),(c). Then take $i = skk\in R$ and $1 = s(ki)\in R_2$. This 1 satisfies 1.4.18.(b),(c). □

As an example of the usefulness and flexibility of the notion of functional lambda model we will show how the *graphmodel* $\mathbb{P}\omega$ can be introduced in this way. Let us first mention some difficulties: since in a functional domain, we have that $[X\to X]$ is embedded into X (by the mapping G which has a leftinverse F), we get by Cantor's theorem that $[X\to X]\neq X^X$, if card $(X)>1$. Here X^X is the full set theoretical function space. Hence we have to restrict the functions in an appropriate way to get $[X\to X]$. A second difficulty is to satisfy the conditions in proposition 1.4.15. to show that we get a functional lambda model. In particular $(X,\cdot)$ should be combinatory complete. Both difficulties can be solved by putting an appropriate topology on X and by taking $[X\to X]$ to consist of all continuous mappings on X. If this topology is well-chosen, we can really embed $[X\to X]$ into X and we can show combinatory completeness by demonstrating that every algebraic function is continuous and every continuous function is representable, so all three classes of functions coincide. Suitable structures on which such a topology can be introduced are the complete partial orders.

1.4.22. Definition. Let $(X,\leqslant)$ be a *partial order* (po).

(i) A subset $D\subseteq X$ is *directed* if $D\neq\emptyset$ and for every $x,y\in D$ there exists $z\in D$ such that $x,y\leqslant z$.

(ii) $(X,\leqslant,\perp)$ is a *complete partial order* (cpo) if $\perp\in X$ is the least element of X and every directed set $D\subseteq X$ has a supremum $\sup D\in X$.

□

1.4.23. Definition (the Scott topology).

Let $(X,\leqslant,\perp)$ be a cpo. $U\subseteq X$ is *Scott open* if

(a) $\forall x,y\in X(x\in U\wedge x\leqslant y\rightarrow y\in U)$,

(b) $\forall D\subseteq X\,(D \text{ directed}\wedge \sup D\in U\rightarrow D\cap U\neq\emptyset)$.

□

1.4.24. Lemma. Let $(X,\leqslant,\perp)$ and $(X',\leqslant',\perp')$ be cpo's.

(i) $\{U\subseteq X\mid U \text{ is Scott open}\}$ defines a topology on X.

(ii) $f\colon X\rightarrow X'$ is continuous $\Leftrightarrow$ $\forall D\subseteq X\,(D \text{ directed}\rightarrow f(\sup D)=\sup f(D))$.

Proof:

(i) Easy.

(ii) Note that continuous functions are monotonic and use the fact that for any $y\in X'$ $\{z\mid z\not\leqslant' y\}$ is open.

□

From now on we will write X for $(X,\leqslant,\perp)$, making no notational distinction between the partial order relations and bottom elements of the different cpo's.

1.4.25. Definition. Let X,Y be cpo's.

(i) Define a partial ordering on X×Y by
$\langle x_1,y_1\rangle\leqslant\langle x_2,y_2\rangle \Leftrightarrow x_1\leqslant x_2\wedge y_1\leqslant y_2$.

(ii) Define a partial ordering on $[X\rightarrow Y]=\{f\colon X\rightarrow Y\mid f \text{ is continuous}\}$ by $f\leqslant g \Leftrightarrow \forall x\in X(f(x)\leqslant g(x))$.

□

1.4.26. Lemma. Let X,Y be cpo's.

(i) $X \times Y$ is a cpo.

(ii) $[X \to Y]$ is a cpo.

Proof:

(i) Take $\sup_{X\times Y} D = (\sup_X D_0, \sup_Y D_1)$, where

$D_0 = \{x \in X \mid \exists y \in Y \ \langle x,y\rangle \in D\}$ and

$D_1 = \{y \in Y \mid \exists x \in X \ \langle x,y\rangle \in D\}$.

(ii) Take $\sup_{[X\to Y]} F = (x \mapsto \sup_Y \{f(x) \mid f \in F\})$. □

1.4.27. Lemma. Let X,Y,Z be cpo's.

$f\colon X\times Y \to Z$ is continuous $\Longleftrightarrow$ f is continuous is both arguments separately.

Proof: Easy, using lemma 1.4.24.(ii). □

1.4.28. Definition. Let X,Y,Z be cpo's.

(i) $ev\colon [X\to Y]\times X \to Y$ is defined by

$ev(f,x) = f(x)$.

(ii) $\Lambda\colon [Z\times X \to Y] \to [Z \to [X \to Y]]$ is defined by

$\Lambda(g) = (z \mapsto (x \mapsto g(z,x)))$. □

1.4.29. Lemma. Let X,Y,Z be cpo's.

(i) $ev\colon [X\to Y]\times X \to Y$ is continuous.

(ii) $\Lambda(g)$ is welldefined for any $g \in [Z\times X \to Y]$ and Λ itself is continuous.

Proof: Routine. □

1.4.30. Theorem. Let CPO be the category of cpo's with continuous maps as morphisms. Then CPO is a cartesian closed category.

Proof: Routine. □

1.4.31. Example (the graphmodel, Plotkin [1972], Scott [1974]).

$P\omega = \{X \mid X \subseteq \omega\}$ partially ordered by inclusion is a cpo with $\emptyset$ as bottom element. Let $(e_n)_{n\in\omega}$ be a standard coding of finite subsets of ω and $(n,m) \mapsto \langle n,m\rangle$ be a standard coding of pairs of natural numbers. Define

$$A\cdot B = \{m \in \omega \mid \exists n(e_n \subseteq B \wedge \langle n,m\rangle \in A)\}.$$

It is easy to see that $\cdot : P\omega^2 \to P\omega$ is continuous.
By lemma 1.4.29.(ii) $\Lambda(\cdot) = F: P\omega \to [P\omega \to P\omega]$ is continuous. Define

$$G: [P\omega \to P\omega] \to P\omega \quad \text{by}$$
$$G(f) = \{\langle n,m\rangle \mid m \in f(e_n)\}.$$

Again it is clear that G is continuous.

Trivially every representable function is algebraic and, since $\cdot$ is continuous, every algebraic function is continuous. That every continuous function is representable and hence that the three concepts coincide in this case, follows from the following fact:

$$F\circ G = id_{[P\omega \to P\omega]}.$$

For let $f: P\omega \to P\omega$ be continuous. Then:

$$\begin{aligned} F(G(f))(A) &= G(f)\cdot A \\ &= \{m \mid \exists e_n \subseteq A\ (\langle n,m\rangle \in G(f))\} \\ &= \{m \mid \exists e_n \subseteq A\ (m \in f(e_n))\} \\ &= \{m \mid m \in \cup \{f(e_n) \mid e_n \subseteq A\}\} \\ &= \{m \mid m \in f(A)\}, \text{ by continuity of f,} \\ &= f(A). \end{aligned}$$

We may conclude that $\mathbb{P}\omega = (P\omega,\cdot,F,G)$ is a functional domain. Moreover $(P\omega,\cdot)$ is combinatory complete, since the concepts of representability and algebraicity coincide with the concept of continuity. Since F,G are continuous, so is $G\circ F$; hence $G\circ F$ is representable. By proposition 1.4.15. it follows that $\mathbb{P}\omega$ is a lambda model, the so-called *graphmodel*.

□

This is the first example where we see that lambda calculus can be interpreted in a cartesian closed category. In this case the category, CPO, is concrete or set-like which makes the resulting structure a lambda model. In general, as will be seen in chapter 2, we will get all lambda algebras in a natural way by interpreting in more general cartesian closed categories.

1.5. *Extensionality.*

Loosely speaking we can say the following. In combinatory algebras every algebraic function is representable. In lambda algebras this representation of algebraic functions can be given uniformly (by the interpretation of lambda terms). In lambda models there is even a canonical representative for every representable function, the association of a canonical representative to any representable function being representable itself.
In this section we will study the ultimate structures in this hierarchy, the extensional lambda models, in which every representable function has a unique representative.

1.5.1. Definition. Let $(X,\circ)$ be an applicative structure.
$(X,\circ)$ is called *extensional* if for all $a,b \in X$

$$\forall x \in X(ax = bx) \to a = b.$$

□

The following proposition shows that extensionality is a very strong property.

1.5.2. Proposition. Let $(X,\circ)$ be an extensional combinatory complete applicative structure. Then there exists a unique pair $(s,k) \in X^2$ such that $(X,\circ,s,k)$ is a combinatory algebra. Moreover this is a lambda model.

Proof: Unicity follows because we have by induction on n:

If $\vec{x} = x_1,\dots,x_n$ then for all $a,b \in X$
$\forall\vec{x}(a\vec{x} = b\vec{x}) \to a = b$.

Hence $\forall xyz(s_1xyz = s_2xyz) \to s_1 = s_2$ and similarly for k.
Existence follows because of combinatory completeness. Let $(X,\circ,s,k)$ be this combinatory algebra. Then $(X,\circ,i)$ is a Meyer lambda model

as one easily checks. Hence $(X,\cdot,s,k)$ is a lambda model.

□

Of course we call a structure $\mathfrak{M} = (X,\cdot,\ldots)$ extensional if the corresponding applicative structure $(X,\cdot)$ is extensional. The property of extensionality can be characterized as follows in a combinatory algebra.

1.5.3. Proposition. Let $\mathfrak{M} = (X,\cdot,s,k)$ be a combinatory algebra. Then $\mathfrak{M}$ is extensional $\Longleftrightarrow$ $\mathfrak{M}$ is weakly extensional and $\mathfrak{M} \models 1 = I$.

Proof:

$\Rightarrow$: Suppose $\mathfrak{M} \models P = Q, P,Q \in \mathcal{C}\ell^0(\underline{X},x)$.
Then $\mathfrak{M} \models (\langle x\rangle P)x = (\langle x\rangle Q)x$. By extensionality $\mathfrak{M} \models \langle x\rangle P = \langle x\rangle Q$.
Hence $\mathfrak{M}$ is weakly extensional.

Furthermore $1xy = xy = Ixy$ in $\mathfrak{M}$. Hence by extensionality, applied twice, $\mathfrak{M} \models 1 = I$.

$\Leftarrow$: Suppose $\forall x(ax = bx)$. Then $\mathfrak{M} \models \forall x(\underline{a}x = \underline{b}x)$. By weak extensionality $\mathfrak{M} \models \langle x\rangle.\underline{a}x = \langle x\rangle.\underline{b}x$. That is $1a = 1b$. Since $1 = i$ we get $a = ia = 1a = 1b = ib = b$.

□

In the formulation of functional domains we get the following characterization of extensionality.

1.5.4. Proposition. Let $\mathfrak{M} = (X,\cdot,F,G)$ be a functional domain. Then $\mathfrak{M}$ is extensional $\Longleftrightarrow$ $G\circ F = id_X$. Hence, if $\mathfrak{M}$ is extensional, $\mathfrak{M}$ is a functional lambda model precisely when $\mathfrak{M}$ is combinatory complete.

Proof:

$\Rightarrow$: Suppose $\mathfrak{M}$ is extensional. For all $a \in X$, we have:

$$F\circ G(F(a)) = F(a),$$

hence $G(F(a)).x = a.x$ for all $x \in X$; then by extensionality

$$G(F(a)) = a.$$

$\Leftarrow$: Suppose $G\circ F = id_X$ and $\forall x \in X(a.x = b.x)$. This means $F(a) = F(b)$.
But then $a = G(F(a)) = G(F(b)) = b$.

The rest follows by proposition 1.4.15., since the identity is representable in any combinatory complete structure.

□

1.5.5. Example. $\mathbb{P}\omega$ is not an extensional lambda model. By proposition 1.5.4. it suffices to show that $G \circ F \neq id_{P\omega}$. But for all $A \in P\omega$

$$\begin{aligned} G(F(A)) &= \{\langle n,m\rangle \mid m \in F(A)(e_n)\} \\ &= \{\langle n,m\rangle \mid \exists k(e_k \subseteq e_n \wedge \langle k,m\rangle \in A)\} \supseteq A. \end{aligned}$$

We see that $G \circ F \supseteq id_{P\omega}$, but for most A (for instance if A is finite nonempty) $G(F(A)) \neq A$.

□

1.5.6. Example (construction of an extensional lambda model).
Let D_0 be an arbitrary cpo. By induction on $n \in \omega$ we will define cpo's D_{n+1} and maps $\varphi_n \in [D_n \to D_{n+1}]$, $\gamma_n \in [D_{n+1} \to D_n]$ as follows:

Basis:
$$\begin{aligned} D_1 &= [D_0 \to D_0], \\ \varphi_0(d) &= (e \mapsto d), \\ \gamma_0(f) &= f(\bot). \end{aligned}$$
(Both φ_0 and γ_0 are clearly continuous.)

Inductionstep:
$$\begin{aligned} D_{n+2} &= [D_{n+1} \to D_{n+1}], \\ \varphi_{n+1}(f) &= \varphi_n \circ f \circ \gamma_n, \\ \gamma_{n+1}(g) &= \gamma_n \circ g \circ \varphi_n. \end{aligned}$$

Now consider the following projective system in CPO:

$$\begin{tikzcd} D_0 & D_1 \arrow[l, "\gamma_0"'] & D_2 \arrow[l, "\gamma_1"'] & D_3 \arrow[l, "\gamma_2"'] & \cdots \arrow[l] \end{tikzcd} .$$

Let D_∞ with $(\pi_n : D_\infty \to D_n)$ be a limit cone for this diagram. (In CPO this kind of limit exists.)
Define $\cdot : D_\infty^2 \to D_\infty$ by: for any $x,y \in D_\infty$

$x \cdot y$ = the least element $z \in D_\infty$ such that for all $n \in \omega$
$$\pi_n(z) \geqslant \pi_{n+1}(x)(\pi_n(y)).$$
(Looking carefully at the definition of the limit in CPO one checks that this is welldefined.) The structure $(D_\infty, \cdot)$ turns out to be extensional and combinatory complete. For a detailed account of this, we refer the reader to Barendregt [1981].

□

1.6. *Homomorphisms and embeddings.*

In any situation where one defines some class of structures one wants to know what is the correct corresponding notion of morphism. In the case of lambda algebras and lambda models the correct notion of morphism is most easily recognized in the combinatory versions, since these are quite familiar from universal algebra. Starting with this we will derive equivalent conditions for all the definitional variants we gave.

1.6.1. Definition. Let $(X,\cdot)$ and $(Y,\cdot')$ be two applicative structures.

(i) A map $\varphi\colon X\to Y$ is a *morphism (of applicative structures)* if for all $x,y\in X$

$$\varphi(x.y) = \varphi(x)\cdot'\varphi(y).$$

Let $\mathfrak{M} = (X,\cdot,s,k)$ and $\mathfrak{N} = (Y,\cdot',s',k')$ be two combinatory algebras.

(ii) A map $\varphi\colon X\to Y$ is a *homomorphism* if it is a morphism of applicative structures and moreover $\varphi(s) = s'$, $\varphi(k) = k'$. Notation $\varphi\colon \mathfrak{M}\to\mathfrak{N}$.

(iii) $\varphi\colon \mathfrak{M}\to\mathfrak{N}$ is an *embedding* if φ is injective. Notation $\varphi\colon \mathfrak{M}\hookrightarrow\mathfrak{N}$.

(iv) $\varphi\colon \mathfrak{M}\to\mathfrak{N}$ is an *isomorphism* if there exists a $\psi\colon \mathfrak{N}\to\mathfrak{M}$ such that $\psi\circ\varphi = id_X$ and $\varphi\circ\psi = id_Y$. Notation $\varphi\colon \mathfrak{M}\xrightarrow{\sim}\mathfrak{N}$.

(v) $\mathfrak{M}$ is *embeddable* in $\mathfrak{N}$ if there exists a $\varphi\colon \mathfrak{M}\hookrightarrow\mathfrak{N}$. Notation $\mathfrak{M}\hookrightarrow\mathfrak{N}$.

(vi) $\mathfrak{M}$ is *isomorphic* to $\mathfrak{N}$ if there exists a $\varphi\colon \mathfrak{M}\xrightarrow{\sim}\mathfrak{N}$. Notation $\mathfrak{M}\cong\mathfrak{N}$.

(vii) $\mathfrak{N}$ is a *homomorphic image* of $\mathfrak{M}$ if there exists a $\varphi\colon \mathfrak{M}\to\mathfrak{N}$ with φ surjective as a map, also denoted $\varphi\colon \mathfrak{M}\twoheadrightarrow\mathfrak{N}$. Notation $\mathfrak{M}\twoheadrightarrow\mathfrak{N}$.

□

Definition 1.6.1. fixes the notion of homomorphism for all variants of the definition of lambda algebra or lambda model.

As always in universal algebra we have that bijective homomorphisms are automatically isomorphisms.

1.6.2. Definition. Let $\varphi\colon X\to Y$ be any map.

(i) For $M\in\Lambda(\underline{X})$ we define $M^{\varphi}\in\Lambda(\underline{Y})$ by induction on M's structure as

$$\begin{aligned} \underline{a}^{\varphi} &= \underline{\varphi(a)}\ , \text{if } a \in X,\\ x^{\varphi} &= x\\ (MN)^{\varphi} &= M^{\varphi}N^{\varphi}\\ (\lambda x.M)^{\varphi} &= \lambda x.M^{\varphi}. \end{aligned}$$

(ii) For $P \in \mathcal{C}\ell(\underline{X})$ we define $P^{\varphi} \in \mathcal{C}\ell(\underline{Y})$ inductively as

$$\begin{aligned} \underline{a}^{\varphi} &= \underline{\varphi(a)}\ , \text{if } a \in X,\\ x^{\varphi} &= x,\ S^{\varphi} = S,\ K^{\varphi} = K\\ (PQ)^{\varphi} &= P^{\varphi}Q^{\varphi}. \end{aligned}$$

(Note that it then follows that $(\langle x\rangle P)^{\varphi} = \langle x\rangle P^{\varphi}$.)

□

1.6.3. Lemma. Let $\mathfrak{M} = (X,\cdot,[\![\cdot]\!])$ and $\mathfrak{N} = (Y,\cdot',[\![\cdot]\!]')$ be two lambda algebras. Then $\varphi\colon X \to Y$ is a homomorphism $\iff$

$$\varphi([\![M]\!]) = [\![M^{\varphi}]\!]' \text{ for all } M \in \Lambda^{0}(\underline{X}).$$

Proof:

$\Rightarrow$: By induction on the structure of $P \in \mathcal{C}\ell^{0}(\underline{X})$ it follows that $\varphi([\![P]\!]) = [\![P^{\varphi}]\!]'$. Then

$$\begin{aligned} \varphi([\![M]\!]) &= \varphi([\![M_{CL}]\!])\\ &= [\![M_{CL}^{\varphi}]\!]'\\ &= [\![M^{\varphi}]\!]'. \end{aligned}$$

(Here it is used that $(M^{\varphi})_{CL} = (M_{CL})^{\varphi}$ which is clear by definition 1.6.2.(i),(ii).)

$\Leftarrow$: $\varphi(s) = \varphi([\![S]\!]) = [\![S^{\varphi}]\!]' = [\![S]\!]' = s'$. Similarly $\varphi(k) = k'$. Moreover φ preserves application, since

$$\varphi(a\cdot b) = \varphi([\![\underline{a}\,\underline{b}]\!]) = [\![\underline{\varphi(a)}\ \underline{\varphi(b)}]\!]' = \varphi(a)\cdot'\varphi(b).$$

□

For the definitional variants of Meyer lambda model and functional lambda model the corresponding facts are slightly more complicated. Let me give the case for the Meyer lambda model. From now on, ab may stand for a·b or a·'b, depending on the context.

1.6.4. Proposition. Let $\mathfrak{M} = (X,\cdot,\varepsilon)$ and $\mathfrak{N} = (Y,\cdot',\varepsilon')$ be two Meyer lambda models. A map $\varphi\colon X \to Y$ is a homomorphism $\iff$

φ is a morphism of applicative structures,
$\varphi(\varepsilon) = \varepsilon'$ and moreover there exist $s_0, k_0 \in X$
such that $(X,\cdot,s_0,k_0), (Y,\cdot',\varphi(s_0),\varphi(k_0)) \models CL$.

Proof:

$\Rightarrow$: Let $(X,\cdot,s,k)$ be the combinatory lambda model corresponding to $\mathfrak{M}$ by corollary 1.4.10. Similarly let $(Y,\cdot',s',k')$ correspond to $\mathfrak{N}$. Then we have by definition of homomorphism that $\varphi(s) = s'$ and $\varphi(k) = k'$. Moreover $\varepsilon = s(ki)$ and $\varepsilon' = s'(k'i')$, hence $\varphi(\varepsilon) = \varepsilon'$. Finally $s_0 = s$ and $k_0 = k$ satisfy the last part of the assertion.

$\Leftarrow$: Take $s_0, k_0 \in X$ as given by the assumption above. Then $\mathfrak{M}_0 = (X,\cdot,s_0,k_0,\varepsilon)$ and $\mathfrak{N}_0 = (Y,\cdot',\varphi(s_0),\varphi(k_0),\varepsilon')$ are combinatory algebras with ε. Consider the operation Φ of definition 1.4.8. Let $\Phi(\mathfrak{M}_0) = (X,\cdot,s,k)$ and $\Phi(\mathfrak{N}_0) = (Y,\cdot',s',k')$. These correspond to the original Meyer lambda models $\mathfrak{M}$ and $\mathfrak{N}$, by proposition 1.4.9. and corollary 1.4.10.
By definition of Φ there are terms $P_S(x,y,z)$, $P_K(x,y,z)$ built up using $\cdot$ only such that

$$s = P_S(s_0,k_0,\varepsilon),\ k = P_K(s_0,k_0,\varepsilon) \text{ and}$$
$$s' = P_S(\varphi(s_0),\varphi(k_0),\varepsilon'),\ k' = P_K(\varphi(s_0),\varphi(k_0),\varepsilon').$$

Making use of the fact that $\varphi(\varepsilon) = \varepsilon'$ we conclude that $\varphi(s) = s'$ and $\varphi(k) = k'$.

□

That the strange condition on existence of $s_0, k_0 \in X$ in proposition 1.6.4. is really necessary is part of the following lemma.

1.6.5. <u>Lemma</u>. Let $\mathfrak{M} = (X,\cdot,\varepsilon)$ and $\mathfrak{N} = (Y,\cdot',\varepsilon')$ be two Meyer lambda models.

(i) If $\text{card}(Y) > 1$ then there exists a map $\varphi\colon X \to Y$ preserving $\cdot$ and ε such that φ is not a homomorphism.

(ii) If $\varphi\colon X \to Y$ is surjective and preserves $\cdot$ and ε, then φ is a homomorphism.

Proof:

(i) Consider the map φ, defined by $\varphi(x) = \varepsilon'$ for all $x \in X$.

Then $\varphi(xy) = \varepsilon' = \varepsilon'\varepsilon' = \varphi(x)\varphi(y)$
and $\varphi(\varepsilon) = \varepsilon'$.
Now assume $\varphi(k) = \varepsilon'$ satisfies the K-axiom in $\mathfrak{N}$. Then we have in $\mathfrak{N}$, that for all $y \in Y$ $y = iy = \varepsilon' iy = i$, hence $\mathrm{card}(Y) = 1$, quod non.

(ii) Take any $s_0, k_0 \in X$, such that $(X, \cdot, s_0, k_0) \models CL$. Let $d, e \in Y$. Since φ is surjective there are $a, b \in X$ such that $\varphi(a) = d$ and $\varphi(b) = e$. Then $\varphi(k_0)de = \varphi(k_0)\varphi(a)\varphi(b) = \varphi(k_0 ab) = \varphi(a) = d$. Hence $\varphi(k_0)$ satisfies the K-axiom in $\mathfrak{N}$. Similarly $\varphi(s_0)$ satisfies the S-axiom.

□

Our next goal is to show that every lambda algebra can be represented as a retraction of a lambda model, a result due to Barendregt and Koymans [1980] and Meyer [1981], [1982].

1.6.6. Theorem. Let $\mathfrak{M}$ be a lambda algebra. Then there exists a lambda model $\mathfrak{N}$ such that $\mathfrak{M} \overset{\hookrightarrow}{\twoheadleftarrow} \mathfrak{N}$.

Proof: Let $T = \mathrm{Th}(\mathfrak{M}) = \{M = N \mid M, N \in \Lambda^0(\mathfrak{M}) \wedge \mathfrak{M} \models M = N\}$. This T is a $\lambda(\mathfrak{M})$-theory. Consider $\mathfrak{N} = \mathfrak{M}(T)$. By proposition 1.1.19. $\mathfrak{N}$ is a lambda model.

Define $\varphi : |\mathfrak{M}| \to |\mathfrak{N}|$ by $\varphi(a) = [\underline{a}]_T$
and $\psi_{a_0} : |\mathfrak{N}| \to |\mathfrak{M}|$ by $\psi([M]_T) = [\![M^{a_0}]\!]^{\mathfrak{M}}$,

where a_0 is any element of $\mathfrak{M}$ and M^{a_0} is M with all free variables replaced by $\underline{a_0}$.

It should be clear that φ and ψ are homomorphisms and moreover that $\psi_{a_0}(\varphi(a)) = a$ for all $a \in \mathfrak{M}$.

□

As was shown in Barendregt and Koymans [1980] a similar correspondence between lambda models and extensional lambda models does not hold.

1.6.7. Proposition. There exists a lambda model $\mathfrak{M}$ such that

(a) for no extensional lambda model $\mathfrak{N}$ we have $\mathfrak{M} \hookrightarrow \mathfrak{N}$ or $\mathfrak{N} \twoheadrightarrow \mathfrak{M}$,

(b) for no nontrivial extensional lambda model $\mathfrak{N}'$ we have $\mathfrak{M} \twoheadrightarrow \mathfrak{N}'$.

Proof: See Barendregt and Koymans [1980], corollary 4.11. The fact that $\mathfrak{N} \twoheadrightarrow \mathfrak{M}$ is impossible follows because $\mathfrak{N} \models 1 = I \Rightarrow \mathfrak{M} \models 1 = I$, which does not hold for the $\mathfrak{M}$, constructed in this paper.

□

In proposition 1.6.7. with nontrivial we mean a structure $(X, \cdot, \ldots)$ such that $\text{card}(X) > 1$. Clearly the singleton structure is formally an extensional lambda model. But in almost all considerations from now on this trivial model will be excluded without mention. If we do this we can state the following theorem which tells us that recursion theoretically and algebraically speaking combinatory algebras are not so trivial.

1.6.8. Theorem. Combinatory algebras are

(a) never commutative,

(b) never associative,

(c) never recursive (hence never finite).

Proof:

(a) If $ik = ki$, then for all x,y
$x = kxy = ikxy = kixy = iy = y$.

(b) If $kii = k(ii)$, then for all x
$x = ix = kiix = k(ii)x = ii = i$.

(c) This follows from the fact that CL is essentially undecidable.

□

CHAPTER 2

CATEGORICAL DEFINITIONS OF THE MODELS

In this chapter we want to investigate in more detail in how far lambda calculus can be conceived of as a theory of functions. In 1.4.14. we saw that if one considers functions in the traditional set theoretic way we end up with the definition of a lambda model. These structures are weakly extensional, which is not surprising since set theoretic functions are given by their input-output behaviour. But as we saw in chapter 1 there are legitimate structures for interpreting lambda calculus that do not satisfy weak extensionality. We saw essentially two different ways to arrive at these socalled lambda algebras, the crude and the combinatorial approach. What we want to show in this chapter is how we can arrive at the definition of lambda algebras, but still making use of the function-concept. It is clear that we have to choose a more intensional view of functions and hence nothing seems more appropriate than a category theoretic approach. It turns out that this is both completely natural and completely general. The idea of interpreting lambda calculus in certain categories goes back to Scott [1980].

2.1. *Categorical models.*

The following definitions introduce some notation that will be used throughout the chapter.

2.1.1. Definition.

(i) C is a cartesian closed category (c.c.c.), that is C is a category such that

(a) there exists a terminal object $T \in |C|$ satisfying for all $A \in |C|$ there exists a unique $!_A \in C(A,T)$;

(b) for every $A,B \in |C|$ there exists a product $A\times B \in |C|$ and maps $p_{AB} \in C(A\times B,A)$, $q_{AB} \in C(A\times B,B)$ such that for every $C \in |C|$ and $f \in C(C,A)$, $g \in C(C,B)$ there exists a unique $\langle f,g\rangle \in C(C,A\times B)$ satisfying

$$\begin{tikzcd}
& A \\
C \arrow[ur, "f"] \arrow[r, "\langle f{,}g\rangle"] \arrow[dr, "g"'] & A\times B \arrow[u, "p_{AB}"'] \arrow[d, "q_{AB}"] \\
& B
\end{tikzcd}$$

$$f = p_{AB} \circ \langle f,g\rangle, \quad g = q_{AB} \circ \langle f,g\rangle.$$

For $f \in C(A,B)$, $g \in C(C,D)$ define
$f\times g \in C(A\times C,B\times D)$ by
$f\times g = \langle f \circ p_{AC}, g \circ q_{AC}\rangle$;

(c) for every $A,B \in |C|$ there exists a function space $B^A \in |C|$ and a map $ev_{AB}: B^A\times A \to B$ such that for every $C \in |C|$ and every $f \in C(C\times A,B)$ there exists a unique $\Lambda_{ABC}(f) \in C(C,B^A)$ satisfying

$$\begin{tikzcd}
B^A\times A \arrow[r, "ev_{AB}"] & B \\
C\times A \arrow[u, "\Lambda_{ABC}(f)\times id_A"] \arrow[ur, "f"'] &
\end{tikzcd}$$

$$ev_{AB} \circ \Lambda_{ABC}(f)\times id_A = f.$$

(ii) For $A \in |C|$, define $A^n \in |C|$ by induction on $n \in \omega$ as

$$A^0 = T\ ,\ A^{n+1} = A^n\times A.$$

Note that $A^1 \equiv T\times A \cong A$, but $A^1 \not\equiv A$.

(iii) Define by induction on $n \in \omega$, for maps $f_i: A \to A_i$, $1 \leqslant i \leqslant n$,

$[f_1,\dots,f_n]\colon A \to T\times A_1 \times \dots \times A_n$ as

$$[\;\;] = !_A\ ,$$

$$[f_1,\dots,f_n, f_{n+1}] = \langle [f_1,\dots,f_n], f_{n+1}\rangle .$$

Note that × associates to the left.

□

Whenever no confusion arises we leave out the subscripts of ev, Λ , $!$, p , q , ... Now let us note the following equations.

$$\langle f,g\rangle \circ h = \langle f\circ h,\ g\circ h\rangle \qquad (1)$$

$$[f_1,\dots,f_n]\circ h = [f_1\circ h,\dots,f_n\circ h] \qquad (2)$$

$$f\times g\circ\langle h,k\rangle = \langle f\circ h,\ g\circ k\rangle \qquad (3)$$

$$\Lambda(h\circ g\times id) = \Lambda(h)\circ g \qquad (4).$$

In order to interpret lambda calculus we need an object U and maps F,G in analogy with the functional modeldefinition 1.4.11.

2.1.2. Definition.

(i) Let $U\in |C|$, $F\in C(U,U^U)$, $G\in C(U^U,U)$, such that F is an epimorphism. Then we call (C,U,F,G) a *categorical lambda structure*.

(ii) U is called a *reflexive object* if $F\circ G = id_{U^U}$. In this case (C,U,F,G) is called a *categorical lambda algebra*. Then we have a retraction $(G,F)\colon U^U \lhd U$:

$$\begin{tikzcd} U^U \arrow[r, shift left, "G"] & U \arrow[l, shift left, "F"] \end{tikzcd} .$$

□

The fact that F is an epimorphism tells us that the function space U^U in C is really the "set of all representable functions on U".

As a last preparation we specify how to handle sequences of distinct variables.

2.1.3. Definition. Let $\Gamma,\Delta,\Theta,\dots$ denote finite sequences of distinct variables; $\langle\;\rangle$ denotes the empty sequence.

(i) If $\Delta = x_1,\ldots,x_n$ then

$\#\Delta = n$,

$X\Delta = U^{\#\Delta}$,

$|\Delta| = \{x_1,\ldots,x_n\}$,

$\Delta\backslash x = x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n$, if $x \equiv x_i$,

$= \Delta$, if $x \notin |\Delta|$,

$\Delta;x = \Delta\backslash x,x$.

(ii) If $\Delta = x_1,\ldots,x_n$ and $1 \leqslant i \leqslant n$ then $\pi^{\Delta}_{x_i} : X\Delta \to U$ is defined by induction on n.

$\pi^{\Delta}_{x_i} = q$, if $i = n$,

$= \pi^{\Delta\backslash x_n}_{x_i} \circ p$, if $i \neq n$.

Here $q\colon U^{n-1}\times U \to U$, $p\colon U^{n-1}\times U \to U^{n-1}$.

Think of $\pi^{\Delta}_{x_i}$ as the mapping: $(a_1,\ldots,a_n) \mapsto a_i$.

(iii) Let $\Gamma = y_1,\ldots,y_m$ with $|\Gamma| \subseteq |\Delta|$. Define

$\Pi^{\Delta}_{\Gamma}\colon X\Delta \to X\Gamma$ by

$\Pi^{\Delta}_{\Gamma} = [\pi^{\Delta}_{y_1},\ldots,\pi^{\Delta}_{y_m}]$.

If $y_i = x_{\varphi(i)}$, $1 \leqslant i \leqslant m$, then think of the mapping

$\Pi^{\Delta}_{\Gamma}\colon (a_1,\ldots,a_n) \mapsto (a_{\varphi(1)},\ldots,a_{\varphi(m)})$. □

The following equations can be easily derived from these definitions, where $\Delta = x_1,\ldots,x_n$, $|\Theta| \subseteq |\Gamma| \subseteq |\Delta|$, $y \in |\Gamma|$, $x \notin |\Delta|$ and $i \in \{1,\ldots,n\}$.

$$\pi^{\Delta}_{x_i} \circ [f_1,\ldots,f_n] = f_i \tag{5}$$

$$\Pi^{\Delta}_{x_i} \circ [f_1,\ldots,f_n] = [f_i] = \langle\, !\, , f_i \rangle \tag{6}$$

$$\pi^{\Gamma}_{y} \circ \Pi^{\Delta}_{\Gamma} = \pi^{\Delta}_{y} \tag{7}$$

$$\Pi^{\Gamma}_{\Theta} \circ \Pi^{\Delta}_{\Gamma} = \Pi^{\Delta}_{\Theta} \tag{8}$$

$$\Pi^{\Delta,x}_{\Gamma,x} = \Pi^{\Delta}_{\Gamma} \times \mathrm{id}_U \tag{9}$$

$$\Pi_{\Gamma}^{\Delta,x} = \Pi_{\Gamma}^{\Delta} \circ p \qquad (10)$$

$$\Pi_{\Delta}^{\Delta} = id_{X\Delta} \qquad (11).$$

Now, at last, we can define the interpretation of lambda calculus with respect to the categorical lambda structure (C,U,F,G). Sometimes we will write C in stead of (C,U,F,G).

2.1.4. Definition.

(i) For any $A \in |C|$ let $\bullet_A : C(A,U)^2 \to C(A,U)$ be defined by

$$f \bullet_A g = ev \circ \langle F \circ f, g \rangle.$$

(ii) Let $\mathfrak{M} = (X, \bullet)$ be the applicative structure with

$$X = C(T,U) \quad \text{and} \quad \bullet = \bullet_T .$$

(iii) Define, for $M \in \Lambda(\mathfrak{M})$ and $|\Delta| \supseteq FV(M)$, $[\![M]\!]_\Delta : X\Delta \to U$ by induction on M as follows.

$$[\![x]\!]_\Delta = \pi_x^\Delta$$

$$[\![\underline{a}]\!]_\Delta = a \circ \Pi_{\langle\ \rangle}^{\Delta}$$

$$[\![MN]\!]_\Delta = [\![M]\!]_\Delta \bullet_{X\Delta} [\![N]\!]_\Delta$$

$$[\![\lambda x.M]\!]_\Delta = G \circ \Lambda([\![M]\!]_{\Delta;x}) \circ \Pi_{\Delta\backslash x}^{\Delta}.$$

(iv) Let $\mathfrak{M}(C) = (X, \bullet, [\![\cdot]\!])$, where $[\![\cdot]\!] = [\![\cdot]\!]_{\langle\ \rangle}$. □

Finally we note the following equation, for any h: $B \to A$ and $f,g\colon A \to U$.

$$(f \bullet_A g) \circ h = (f \circ h) \bullet_B (g \circ h) \qquad (12).$$

2.1.5. Lemma.

(i) If $|\Delta| \supseteq |\Gamma| \supseteq FV(M)$, then

$$[\![M]\!]_\Delta = [\![M]\!]_\Gamma \circ \Pi_\Gamma^\Delta.$$

(ii) Let $|\Delta| = \{\vec{x}\} \supseteq FV(M)$ and $|\Gamma| \supseteq FV(\vec{N})$. Then

$$[\![M[\vec{x} := \vec{N}]]\!]_\Gamma = [\![M]\!]_\Delta \circ [[\![\vec{N}]\!]_\Gamma].$$

Proof: The properties are intuitively clear if one thinks of the heuristic meaning: $\llbracket M(\vec{x})\rrbracket_\Delta = $ "$\vec{a} \mapsto M(\vec{\underline{a}})$".

(i) By induction on $M \in \Lambda(\mathfrak{M})$.

$$\begin{aligned}
\llbracket x\rrbracket_\Delta &= \pi^\Delta_x \\
&= \pi^\Gamma_x \circ \Pi^\Delta_\Gamma && \text{, by (7),} \\
&= \llbracket x\rrbracket_\Gamma \circ \Pi^\Delta_\Gamma .
\end{aligned}$$

$$\begin{aligned}
\llbracket \underline{a}\rrbracket_\Delta &= a \circ \Pi^\Delta_{\langle\,\rangle} \\
&= a \circ \Pi^\Gamma_{\langle\,\rangle} \circ \Pi^\Delta_\Gamma && \text{, by (8),} \\
&= \llbracket \underline{a}\rrbracket_\Gamma \circ \Pi^\Delta_\Gamma .
\end{aligned}$$

$$\begin{aligned}
\llbracket MN\rrbracket_\Delta &= \llbracket M\rrbracket_\Delta \bullet_{X\Delta} \llbracket N\rrbracket_\Delta \\
&= (\llbracket M\rrbracket_\Gamma \circ \Pi^\Delta_\Gamma) \bullet_{X\Delta} (\llbracket N\rrbracket_\Gamma \circ \Pi^\Delta_\Gamma) && \text{, by (IH),} \\
&= (\llbracket M\rrbracket_\Gamma \bullet_{X\Gamma} \llbracket N\rrbracket_\Gamma) \circ \Pi^\Delta_\Gamma && \text{, by (12),} \\
&= \llbracket MN\rrbracket_\Gamma \circ \Pi^\Delta_\Gamma .
\end{aligned}$$

$$\begin{aligned}
\llbracket \lambda x.M\rrbracket_\Delta &= G \circ \Lambda(\llbracket M\rrbracket_{\Delta;x}) \circ \Pi^\Delta_{\Delta\backslash x} \\
&= G \circ \Lambda(\llbracket M\rrbracket_{\Gamma;x} \circ \Pi^{\Delta;x}_{\Gamma;x}) \circ \Pi^\Delta_{\Delta\backslash x} && \text{, by (IH),} \\
&= G \circ \Lambda(\llbracket M\rrbracket_{\Gamma;x} \circ \Pi^{\Delta\backslash x}_{\Gamma\backslash x} \times id_U) \circ \Pi^\Delta_{\Delta\backslash x} && \text{, by (9),} \\
&= G \circ \Lambda(\llbracket M\rrbracket_{\Gamma;x}) \circ \Pi^{\Delta\backslash x}_{\Gamma\backslash x} \circ \Pi^\Delta_{\Delta\backslash x} && \text{, by (4),} \\
&= G \circ \Lambda(\llbracket M\rrbracket_{\Gamma;x}) \circ \Pi^{\Gamma}_{\Gamma\backslash x} \circ \Pi^\Delta_{\Gamma} && \text{, by (8),} \\
&= \llbracket \lambda x.M\rrbracket_\Gamma \circ \Pi^\Delta_\Gamma .
\end{aligned}$$

(ii) By induction on $M \in \Lambda(\mathfrak{M})$.

The only difficult case is when $M \equiv \lambda y.L$. Let $|\Delta'| = FV(M) \subseteq |\Delta|$, say $\Delta' = \vec{x}'$. Let $\vec{N}'$ be the corresponding subsequence of $\vec{N}$. Note that $y \notin |\Delta'|$ and $y \notin FV(\vec{N}')$. (By the conventions on substitution.) Then

$$\begin{aligned}
\llbracket M[\vec{x} := \vec{N}]\rrbracket_{\Gamma} &= \llbracket \lambda y.L[\vec{x}' := \vec{N}',\ y := y]\rrbracket_{\Gamma} \\
&= G \circ \Lambda(\llbracket L[\vec{x}' := \vec{N}',\ y := y]\rrbracket_{\Gamma;y}) \circ \Pi^{\Gamma}_{\Gamma\backslash y} \\
&= G \circ \Lambda(\llbracket L\rrbracket_{\Delta';y} \circ [\llbracket \vec{N}'\rrbracket_{\Gamma;y},\ \llbracket y\rrbracket_{\Gamma;y}]) \circ \Pi^{\Gamma}_{\Gamma\backslash y}, \text{ by (IH),} \\
&= G \circ \Lambda(\llbracket L\rrbracket_{\Delta';y} \circ \langle\, [\llbracket \vec{N}'\rrbracket_{\Gamma\backslash y}] \circ p, q\rangle) \circ \Pi^{\Gamma}_{\Gamma\backslash y}, \text{ by (i), (10),} \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(11),} \\
&= G \circ \Lambda(\llbracket L\rrbracket_{\Delta';y} \circ [\llbracket \vec{N}'\rrbracket_{\Gamma\backslash y}] \times id_U) \circ \Pi^{\Gamma}_{\Gamma\backslash y} \\
&= G \circ \Lambda(\llbracket L\rrbracket_{\Delta';y}) \circ [\llbracket \vec{N}'\rrbracket_{\Gamma\backslash y}] \circ \Pi^{\Gamma}_{\Gamma\backslash y} \quad \text{, by (4),} \\
&= G \circ \Lambda(\llbracket L\rrbracket_{\Delta';y}) \circ [\llbracket \vec{N}'\rrbracket_{\Gamma}] \quad \text{, by (i), (2),} \\
&= G \circ \Lambda(\llbracket L\rrbracket_{\Delta';y}) \circ \Pi^{\Delta}_{\Delta'} \circ [\llbracket \vec{N}\rrbracket_{\Gamma}] \quad \text{, by (5),} \\
&= \llbracket \lambda y.L\rrbracket_{\Delta'} \circ \Pi^{\Delta}_{\Delta'} \circ [\llbracket \vec{N}\rrbracket_{\Gamma}] \quad \text{, since } y \notin |\Delta'|, \\
&= \llbracket \lambda y.L\rrbracket_{\Delta} \circ [\llbracket \vec{N}\rrbracket_{\Gamma}] \quad \text{, by (i).}
\end{aligned}$$

□

2.1.6. Corollary.

(i) If $|\Delta| \supseteq FV(\lambda x.M)$ and $|\Gamma| \supseteq FV(N) \cup |\Delta\backslash x|$ then

$$\llbracket M[x := N]\rrbracket_{\Gamma} = \llbracket M\rrbracket_{\Delta;x} \circ \langle\, \Pi^{\Gamma}_{\Delta\backslash x},\ \llbracket N\rrbracket_{\Gamma}\rangle.$$

(ii) If $|\Delta| \supseteq FV(\lambda x.M)$ and $x,y \notin |\Delta|$ then

$$\llbracket M[x := y]\rrbracket_{\Delta;y} = \llbracket M\rrbracket_{\Delta;x}.$$

Proof:

(i) Let $\Delta\backslash x = \vec{y}$. Then

$$\begin{aligned}
\llbracket M[x := N]\rrbracket_{\Gamma} &= \llbracket M[\vec{y} := \vec{y},\ x := N]\rrbracket_{\Gamma} \\
&= \llbracket M\rrbracket_{\Delta;x} \circ [\llbracket \vec{y}\rrbracket_{\Gamma},\ \llbracket N\rrbracket_{\Gamma}], \text{ by 2.1.5.(ii),} \\
&= \llbracket M\rrbracket_{\Delta;x} \circ \langle\, [\llbracket \vec{y}\rrbracket_{\Gamma}],\ \llbracket N\rrbracket_{\Gamma}\rangle \\
&= \llbracket M\rrbracket_{\Delta;x} \circ \langle\, \Pi^{\Gamma}_{\Delta\backslash x},\ \llbracket N\rrbracket_{\Gamma}\rangle.
\end{aligned}$$

$$\begin{aligned}
\text{(ii) } \llbracket M[x := y]\rrbracket_{\Delta;y} &= \llbracket M\rrbracket_{\Delta;x} \circ \langle\, \Pi^{\Delta;y}_{\Delta\backslash x},\ \llbracket y\rrbracket_{\Delta;y}\rangle, \text{ by (i),} \\
&= \llbracket M\rrbracket_{\Delta;x} \circ \langle\, p,q\rangle, \text{ since } \Delta\backslash x = \Delta = \Delta\backslash y,
\end{aligned}$$

$$= [\![M]\!]_{\Delta;x}, \text{ since } \langle p,q\rangle = id.$$

□

2.1.7. Proposition. Let $(\mathcal{C},U,F,G)$ be a categorical lambda structure. If

$$[\![(\lambda y.xy)y]\!]_{x,y} = [\![xy]\!]_{x,y}$$

then U is a reflexive object with respect to F,G, i.e. $(\mathcal{C},U,F,G)$ is a categorical lambda algebra.

Proof: We may compute as follows.

$$\begin{aligned}
[\![x]\!]_{x,y} &= q \circ p\ ,\\
[\![y]\!]_{x,y} &= q\ ,\\
[\![xy]\!]_{x,y} &= ev \circ \langle F \circ q \circ p, q\rangle = ev \circ (F \circ q) \times id,\\
[\![\lambda y.xy]\!]_{x,y} &= G \circ F \circ q \circ p,\\
[\![(\lambda y.xy)y]\!]_{x,y} &= ev \circ \langle F \circ G \circ F \circ q \circ p, q\rangle = ev \circ (F \circ G \circ F \circ q) \times id.
\end{aligned}$$

Hence we have $ev \circ (F \circ q) \times id = ev \circ (F \circ G \circ F \circ q) \times id$. By applying Λ we get $F \circ q = F \circ G \circ F \circ q$. Now $q : T{\times}U \xrightarrow{\sim} U$ is an isomorphism and F is epi, hence $F \circ G = id$.

□

Here we see that the β-axiom forces U to be a reflexive object. On the other hand this is sufficient for proving soundness of the lambda calculus.

2.1.8. Theorem. Let $(\mathcal{C},U,F,G)$ be a categorical lambda algebra. Assume $\lambda(\mathfrak{M}(\mathcal{C})) \vdash M = N$ and $|\Delta| \supseteq FV(MN)$. Then $[\![M]\!]_\Delta = [\![N]\!]_\Delta$.

Proof: By induction on the length of proof of $M = N$. We need only check the β-axiom because the rest is easy. (In particular the ξ-rule trivially holds, since $[\![M]\!]_\Delta$ gives an intensional interpretation of the function "$\lambda\vec{x}.M$" $\in \mathcal{C}(U^n,U)$.) Therefore we have to show

$$[\![(\lambda x.M)N]\!]_\Delta = [\![M[x := N]]\!]_\Delta, \text{ whenever } |\Delta| \supseteq FV((\lambda x.M)N).$$

Now $$\begin{aligned}
[\![(\lambda x.M)N]\!]_\Delta &= (G \circ \Lambda([\![M]\!]_{\Delta;x}) \circ \Pi^\Delta_{\Delta\backslash x}) \bullet_{X\Delta} [\![N]\!]_\Delta\\
&= ev \circ \langle \Lambda([\![M]\!]_{\Delta;x}) \circ \Pi^\Delta_{\Delta\backslash x}, [\![N]\!]_\Delta\rangle, \text{ since } F \circ G = id,\\
&= ev \circ \Lambda([\![M]\!]_{\Delta;x}) \times id \circ \langle \Pi^\Delta_{\Delta\backslash x}, [\![N]\!]_\Delta\rangle, \text{ by (3)},
\end{aligned}$$

$$= [\![M]\!]_{\Delta;x} \circ \langle \Pi^{\Delta}_{\Delta\setminus x}, [\![N]\!]_{\Delta} \rangle$$

$$= [\![M[x := N]]\!]_{\Delta} \quad \text{, by 2.1.6.(i).}$$

□

2.1.9. Corollary. Let (C,U,F,G) be a categorical lambda algebra. Then $\mathfrak{M}(C)$ is a lambda algebra.

Proof: This is clear from theorem 2.1.8. and lemma 1.1.9.

□

We call $\mathfrak{M}(C)$ the lambda algebra generated by C.

2.1.10. Definition. Let (C,U,F,G) be a categorical lambda algebra and $\rho : \text{Vars} \to |\mathfrak{M}(C)|$ an assignment. Let $\Delta = x_1,\ldots,x_n$. Then define

$$\rho^{\Delta} = [\rho(x_1),\ldots,\rho(x_n)]: T \to X\Delta.$$

□

2.1.11. Lemma. Let (C,U,F,G) be a categorical lambda algebra and $M \in \Lambda(\mathfrak{M}(C))$. Assume $\rho : \text{Vars} \to |\mathfrak{M}(C)|$ and $|\Delta| \supseteq FV(M)$. Then

$$[\![M]\!]_{\rho} = [\![M]\!]_{\Delta} \circ \rho^{\Delta}.$$

Proof: Let $\Delta = x_1,\ldots,x_n = \vec{x}$.

Then $[\![M]\!]_{\rho} = [\![M[\vec{x} := \underline{\rho(\vec{x})}]]\!]$

$$= [\![M]\!]_{\Delta} \circ [[\![\underline{\rho(\vec{x})}]\!]] \quad \text{, by lemma 2.1.5.(ii),}$$

$$= [\![M]\!]_{\Delta} \circ \rho^{\Delta}.$$

□

As the last part of this first section we want to investigate when $\mathfrak{M}(C)$ is (weakly) extensional.

2.1.12. Definition. Let C be a category with terminal object T. Let $A \in |C|$ be any object. Then C has *enough points at* A if

$$\forall f,g \in C(A,A)\,(f \neq g \Rightarrow \exists x \in C(T,A)\,(f \circ x \neq g \circ x)).$$

□

2.1.13. Proposition. Let C be a categorical lambda algebra.

(i) C has enough points at $U \Longleftrightarrow \mathfrak{M}(C)$ is weakly extensional.

(ii) $\mathfrak{M}(C) \models 1 = I \iff G \circ F = id_U$.

Proof:

(i) $\Rightarrow$: Let $M,N \in \Lambda^0(|\mathfrak{M}(C)|, x)$ and assume

$$[\![M(\underline{d})]\!] = [\![N(\underline{d})]\!] \text{ for all } d \in |\mathfrak{M}(C)| = C(T,U).$$

Then, by lemma 2.1.5.(ii), $[\![M]\!]_x \circ [d] = [\![N]\!]_x \circ [d]$ for all $d \in C(T,U)$. Since $U^1 = T{\times}U \cong U$, C has also enough points at U^1. Therefore $[\![M]\!]_x = [\![N]\!]_x$ and hence $[\![\lambda x.M]\!] = [\![\lambda x.N]\!]$.

$\Leftarrow$: It suffices to show that for every $f: U \to U$ there exists an $a: T \to U$ such that

(1) for all $d: T \to U$ $(a{\cdot}d = f \circ d)$

and (2) $[\![ax]\!]_x = f \circ q_{TU}$.

(Since then

$$\begin{aligned} \forall x(f \circ x = g \circ x) &\Rightarrow \forall d(a{\cdot}d = b{\cdot}d) \\ &\Rightarrow [\![\lambda x.ax]\!] = [\![\lambda x.bx]\!] \\ &\Rightarrow [\![ax]\!]_x = [\![bx]\!]_x \\ &\Rightarrow \quad f = g \quad .) \end{aligned}$$

But an easy computation shows that $a = G \circ \Lambda(f \circ q_{TU})$ satisfies (1) and (2) above.

This proof may be seen in a better perspective after reading section 2.4.

(ii) We have by easy calculations $[\![I]\!] = G \circ \Lambda(q)$

and $[\![1]\!] = G \circ \Lambda(G \circ F \circ q)$,

where $q: T{\times}U \to U$.

Hence

$$\begin{aligned} \mathfrak{M}(C) \models 1 = I &\iff G \circ \Lambda(G \circ F \circ q) = G \circ \Lambda(q) \\ &\iff \Lambda(G \circ F \circ q) = \Lambda(q), \text{ since } F \circ G = id, \\ &\iff G \circ F \circ q = q \\ &\iff G \circ F = id \text{ , since } q \text{ is iso.} \end{aligned}$$

□

2.1.14. <u>Corollary</u>. Let C be a categorical lambda algebra, such that $F: U \xrightarrow[=]{\sim} U^U$ and $G = F^{-1}$. Assume moreover that C has enough points at U. Then $\mathfrak{M}(C)$ is an extensional lambda model.

Proof: This is clear from proposition 2.1.13. by proposition 1.5.3.

□

2.2. *The Karoubi construction.*

Let C be a categorical lambda algebra. In the category C there may exist objects that are not at all related to the reflexive object U. But every object in C, that is important for the interpretation of lambda calculus, is in fact a retract of U. To be specific, if we restrict our attention to the sub-cartesian-closed-category of C, generated by the reflexive object U, then every object in there is a retract of U. This will be proved now.

2.2.1. <u>Definition</u>. Let C be a category and $A,B \in |C|$. We say that A is a *retract* of B with *section* $g\colon A \to B$ and *projection* $f\colon B \to A$ if $f \circ g = id_A$. In this case $g\colon A \hookrightarrow B$ is a monomorphism and $f\colon B \twoheadrightarrow A$ is an epimorphism,

$$\begin{tikzcd} A \arrow[r, hook, shift left, "g"] & B. \arrow[l, two heads, shift left, "f"] \end{tikzcd}$$

Notation $(g,f)\colon A \lhd B$, or if g and f don't matter $A \lhd B$.

□

2.2.2. <u>Proposition</u>. Let C be a categorical lambda algebra.

(i) $T, U{\times}U$ and U^U are all retracts of U.

(ii) If A and B are retracts of U then so are $A{\times}B$ and B^A.

Proof:

(i) $T \lhd U$: To prove this it clearly suffices to show that $C(T,U) \neq \emptyset$, since then, if $a \in C(T,U)$, $(a, !_U)\colon T \lhd U$.
But $[\![\lambda x.x]\!] \in C(T,U)$, hence $C(T,U) \neq \emptyset$.

$U^U \lhd U$: This is clear because by definition of reflexive object $(G,F)\colon U^U \lhd U$.

$U{\times}U \lhd U$: Since $U^2 = T{\times}U{\times}U \cong U{\times}U$ it suffices to show that $U^2 \lhd U$.

Now let $g = [\![\lambda z.zxy]\!]_{x,y} : U^2 \to U$,

$f_1 = [\![wK]\!]_w \qquad : U^1 \to U$,

$$f_2 = [\![w(KI)]\!]_w \quad : U^1 \to U.$$

Then

$$\begin{aligned}[\![x]\!]_{x,y} &= [\![(wK)[w := \lambda z.zxy]]\!]_{x,y}\\ &= [\![wK]\!]_w \circ [[\![\lambda z.zxy]\!]_{x,y}] \text{ , by lemma 2.1.5.(ii),}\\ &= f_1 \circ [g].\end{aligned}$$

Similarly $[\![y]\!]_{x,y} = f_2 \circ [g]$.

Defining $f\colon U \to U^2$ by $f = [f_1, f_2] \circ [id]$ we get

$$\begin{aligned}f \circ g &= [f_1, f_2] \circ [g]\\ &= [f_1 \circ [g], f_2 \circ [g]]\\ &= [[\![x]\!]_{x,y}, [\![y]\!]_{x,y}]\\ &= id_{U^2} .\end{aligned}$$

Hence $(g,f)\colon U^2 \lhd U$.

(ii) Suppose A and B are retracts of U,

$$(g_A, f_A)\colon A \lhd U \quad \text{and} \quad (g_B, f_B)\colon B \lhd U.$$

$A{\times}B \lhd U$: Clearly $(g_A{\times}g_B, f_A{\times}f_B)\colon A{\times}B \lhd U{\times}U$. Furthermore by (i) $U{\times}U \lhd U$. Since $\lhd$ is transitive we get the result.

$B^A \lhd U$: Define $g\colon B^A \to U^U$ by

$$g = \Lambda(g_B \circ ev_{AB} \circ id{\times}f_A)$$

and $f\colon U^U \to B^A$ by

$$f = \Lambda(f_B \circ ev_{UU} \circ id{\times}g_A).$$

Then $(g,f)\colon B^A \lhd U^U$, since

$$\begin{aligned}f \circ g &= \Lambda(f_B \circ ev_{UU} \circ id{\times}g_A) \circ g\\ &= \Lambda(f_B \circ ev_{UU} \circ g{\times}id \circ id{\times}g_A), \text{ by (4),}\\ &= \Lambda(f_B \circ g_B \circ ev_{AB} \circ id{\times}f_A \circ id{\times}g_A)\\ &= \Lambda(ev_{AB})\\ &= id_{B^A} .\end{aligned}$$

Moreover by (i) $U^U \lhd U$ and by transitivity of $\lhd$ we have $B^A \lhd U$.

□

This proposition gives us an indication that all important information about the categorical lambda algebra $\mathcal{C}$ lies in the monoid of endomorphisms of U. In fact, morphisms h: $A \to B$, where (g_A, f_A): $A \lhd U$ and (g_B, f_B): $B \lhd U$, can be represented by the endomorphism $h' = g_B \circ h \circ f_A \in \mathcal{C}(U,U)$. Then h can be recovered as $h = f_B \circ h' \circ g_A$. Moreover the objects A, which are retracts of U, can themselves be represented in $\mathcal{C}(U,U)$ as follows: let (g_A, f_A): $A \lhd U$, then $p_A = g_A \circ f_A$ is an idempotent in $\mathcal{C}(U,U)$ that conveys all important information about A in the sense that "$A \cong$ range (p_A)".

The remainder of this section focuses attention on the reconstruction of a category from a given monoid. This will be done in such a way that if the given monoid is cartesian closed (to be explained in the next section), then the corresponding category gives rise to a categorical lambda algebra.

We may consider a monoid as a category with only one object. In accordance with the above observation we will try to embed this category in another one in such a way that in the new category all idempotents split, that is to say that all idempotents can be represented by a retract as indicated above.

Let us now formalize these ideas. We show the constructions not only for categories with one object, but for arbitrary categories.

2.2.3. Definition. Let $\mathcal{A}$ be a category.

(i) a: $A \to A$ is called *idempotent* if

$$a^2 \, (=_{\text{def}} a \circ a) = a,$$

$$\begin{tikzcd} A \arrow[r, "a"] \arrow[rr, bend right, "a"'] & A \arrow[r, "a"] & A . \end{tikzcd}$$

(ii) We say that an idempotent a: $A \to A$ *splits* if there exists an object $B \in |\mathcal{A}|$ and a retraction (r,s): $B \lhd A$ such that $a = r \circ s$

$$\begin{tikzcd} A \arrow[rr, "a"] \arrow[dr, "s"'] & & A \\ & B \arrow[ur, "r"'] & \end{tikzcd} \qquad (s \circ r = id_B).$$

Now suppose R,r,s are mappings defined on all idempotents a: $A \to A$ in $\mathcal{A}$, such that (r_a, s_a): $R_a \lhd A$ represents a, that is $r_a \circ s_a = a$.

We suppose that $R_{id_A} = A$, $r_{id_A} = s_{id_A} = id_A$.

(iii) For idempotents $a,b\colon A \to A$ we define

$$a \subseteq b \iff b \circ a \circ b = a,$$

$$a \restriction b = s_b \circ a \circ r_b.$$

We now can formulate the axiom of partial splitting:

(APS): For all idempotents $a,b\colon A \to A$, whenever $a \subseteq b$, then

$$R_a = R_{a \restriction b},$$

$$r_a = r_b \circ r_{a \restriction b}$$

and $$s_a = s_{a \restriction b} \circ s_b.$$

If (APS) is satisfied we call the expansion (A,R,r,s) a *category with explicit splitting*. When no confusion arises we will write A instead of (A,R,r,s).

(iv) CAT is the category of all categories, where morphisms are functors.

CATES is the category of all categories with explicit splitting, where morphisms are functors that preserve the explicit splitting exactly (not only up to isomorphism).

U : CATES → CAT is the obvious forgetful functor.

□

In order to understand this definition a few remarks are in place. For instance $a \subseteq b$ says that the splitting of b is just a partial splitting of a, in other words the idempotent b has a larger range, R_b, than the idempotent a. Indeed we have

$$a \subseteq b \iff b \circ a \circ b = a \iff r_b \circ (s_b \circ a \circ r_b) \circ s_b = a$$

$$\iff \exists a' : R_b \to R_b (r_b \circ a' \circ s_b = a).$$

$$\begin{tikzcd}
A \arrow[r, "a"] \arrow[d, "s_b"'] & A \\
R_b \arrow[r, "a'"] & R_b \arrow[u, "r_b"']
\end{tikzcd}$$
.

If $a \subseteq b$ then $a \upharpoonright b$ is the unique a' that makes the above diagram commute; $a \upharpoonright b$ is itself an idempotent. Hence $a \upharpoonright b$ is the remaining idempotent if one splits a partially through the retract defined by b. Now the axiom of partial splitting says, that the total splitting of a can be achieved by composing the partial splitting of a through b with the total splitting of the remainder $a \upharpoonright b$:

$$\begin{tikzcd}
A \arrow[rr, "a"] \arrow[d, "s_b"] \arrow[ddr, bend right=70, "s_a"'] & & A \\
R_b \arrow[rr, "a \upharpoonright b"] \arrow[dr, "s_{a \upharpoonright b}"'] & & R_b \arrow[u, "r_b"'] \\
& R_{a \upharpoonright b} \arrow[ur, "r_{a \upharpoonright b}"'] \arrow[uur, bend right=70, "r_a"'] &
\end{tikzcd}$$
.

Now we define a functor $\mathcal{K}$: CAT → CATES (the socalled Karoubi-functor) that embeds a given category into a category with explicit splitting by adding just enough retracts for all idempotents, while not making any unnecessary identifications.

This can be stated more precisely in the language of adjoint functors. For the theory of adjoint functors we refer the reader to MacLane [1971].

2.2.4. Theorem. There exists a functor $\mathcal{K}$: CAT → CATES such that $\mathcal{K} \dashv \mathcal{U}$. Moreover the unit η of this adjunction defines for each $\mathcal{A} \in |\mathrm{CAT}|$ a full and faithful embedding $\eta_{\mathcal{A}}: \mathcal{A} \to \mathcal{UK}(\mathcal{A})$.

Proof:

(a) Definition of $\mathcal{K}$: Let $\mathcal{A} \in |\mathrm{CAT}|$ and define

$|\mathcal{K}(\mathcal{A})| = \{(A,a) \mid a: A \to A \text{ is an idempotent in } \mathcal{A}\},$

$$\mathcal{K}(\mathcal{A})((A,a),(B,b)) = \{f \in \mathcal{A}(A,B) \mid b \circ f \circ a = f\}.$$

Note that the condition $b \circ f \circ a = f$ is equivalent to the conjunction of $b \circ f = f$ and $f = f \circ a$. For, if $b \circ f \circ a = f$, then $b \circ f = b \circ b \circ f \circ a = b \circ f \circ a = f$ and similarly $f \circ a = f$. The other way around, if $b \circ f = f = f \circ a$, then $b \circ f \circ a = f \circ a = f$.

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "a"'] \arrow[dr, "f"] & B \arrow[d, "b"] \\
A \arrow[r, "f"'] & B
\end{tikzcd} .$$

It is clear that this gives a category where

$$\mathrm{id}_{(A,a)} = a,$$

$$((B,b) \xrightarrow{g} (C,c)) \circ ((A,a) \xrightarrow{f} (B,b)) = ((A,a) \xrightarrow{g \circ f} (C,c)).$$

We still have to define the explicit splitting on $\mathcal{K}(\mathcal{A})$. Let $f\colon (A,a) \to (A,a)$ be an idempotent, that is $a \circ f = f \circ a = f = f \circ f$. Then $R_f = (A,f)$, $s_f = f$, $r_f = f$,

$$\begin{tikzcd}
A \arrow[rr, "f"] \arrow[dr, "f"'] \arrow[dd, "a"'] & & A \arrow[dd, "a"] \\
& A \arrow[ur, "f"'] \arrow[dd, "f"] & \\
A \arrow[rr, "f"] \arrow[dr, "f"'] & & A \\
& A \arrow[ur, "f"'] &
\end{tikzcd} .$$

Now the conditions for a category with explicit splitting as formulated in definition 2.2.3. are all satisfied, as one easily checks.

(b) Definition of the unit η (this natural transformation defines for every $\mathcal{A}$ a universal arrow from $\mathcal{A}$ to $\mathcal{UK}(\mathcal{A})$, for the forgetful functor $\mathcal{U}$: CATES → CAT, see MacLane [1971], chapter 4.):

Let $\eta_{\mathcal{A}}\colon \mathcal{A} \to \mathcal{UK}(\mathcal{A})$ be the functor defined by

$$\eta_{\mathcal{A}}(A) = (A, id_A),$$

$$\eta_{\mathcal{A}}(A \xrightarrow{f} B) = (\eta_{\mathcal{A}}(A) \xrightarrow{f} \eta_{\mathcal{A}}(B)).$$

It is clear that $\eta_{\mathcal{A}}$ is a full and faithful functor for all $\mathcal{A} \in |\mathrm{CAT}|$.

Now we want to show the following universal property:

For any $\mathcal{B} \in |\mathrm{CATES}|$ and any functor $H\colon \mathcal{A} \to \mathcal{UB}$ there exists a unique (explicit splitting preserving) $\bar{H}\colon \mathcal{K}(\mathcal{A}) \to \mathcal{B}$ such that $H = \mathcal{U}\bar{H} \circ \eta_{\mathcal{A}}$,

$$\begin{tikzcd}
\mathcal{A} \arrow[r, "\eta_{\mathcal{A}}"] \arrow[dr, "H"'] & \mathcal{UK}(\mathcal{A}) \arrow[d, dashed, "\mathcal{U}\bar{H}"] \\
 & \mathcal{UB}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{K}(\mathcal{A}) \arrow[d, dashed, "\exists!\,\bar{H}"'] \\
\mathcal{B}
\end{tikzcd} .$$

Thus let $\mathcal{B}$ and H be given.

(c) Uniqueness of $\bar{H}$: Suppose $\bar{H}$ satisfies the condition above. Look at the following diagram (a: A → A idempotent)

$$\begin{tikzcd}
A \arrow[rr, "a"] \arrow[dr, "a"'] \arrow[dd, "id_A"'] & & A \arrow[dd, "id_A"] \\
 & A \arrow[ur, "a"'] \arrow[dd, "a"] & \\
A \arrow[rr, "a"] \arrow[dr, "a"'] & & A \\
 & A \arrow[ur, "a"'] &
\end{tikzcd} .$$

Since $\bar{H}$ preserves explicit splitting and because we know by assumption that $\bar{H}((A, id_A)) = HA$ and also $\bar{H}((A, id_A) \xrightarrow{a} (A, id_A)) = H(A \xrightarrow{a} A)$, we get the following explicit splitting in $\mathcal{B}$

$$\begin{tikzcd}
HA \arrow[rr, "Ha"] \arrow[dr, "s_{Ha}"'] & & HA \\
& R_{Ha} \arrow[ur, "r_{Ha}"'] &
\end{tikzcd}$$
.

We may conclude that $\bar{H}((A,a)) = R_{Ha}$ (*) ,

$\bar{H}((A,id_A) \xrightarrow{a} (A,a)) = s_{Ha}$,

$\bar{H}((A,a) \xrightarrow{a} (A,id_A)) = r_{Ha}$.

Now let $(A,a) \xrightarrow{f} (B,b)$ be given. Then $b \circ f \circ a = f$,

$$\begin{tikzcd}
(A,a) \arrow[r, "a"] \arrow[rrr, bend right, "f"'] & (A,id_A) \arrow[r, "f"] & (B,id_B) \arrow[r, "b"] & (B,b)
\end{tikzcd}$$
.

Hence $\bar{H}((A,a) \xrightarrow{f} (B,b)) = \bar{H}((B,id_B) \xrightarrow{b} (B,b)) \circ$

$\bar{H}((A,id_A) \xrightarrow{f} (B,id_B)) \circ \bar{H}((A,a) \xrightarrow{a} (A,id_A))$

$= s_{Hb} \circ Hf \circ r_{Ha}$ (**).

We conclude that $\bar{H}$ is uniquely determined.

(d) Existence of $\bar{H}$: Take (*) and (**) as definitions of $\bar{H}$. Then

$\bar{H}((A,id_A)) = R_{Hid_A} = R_{id_{HA}} = HA$ and

$\bar{H}((A,id_A) \xrightarrow{f} (B,id_B)) = s_{Hid_B} \circ Hf \circ r_{Hid_A}$

$= id_{HB} \circ Hf \circ id_{HA}$

$= Hf.$

Hence $\mathcal{U}\bar{H} \circ \eta_A = H$.

Finally we need to check that $\bar{H}$ is indeed a functor and that it preserves explicit splitting.

(d1) $\bar{H}$ is a functor:

$$\begin{aligned}
\bar{H}(id_{(A,a)}) &= \bar{H}((A,a) \xrightarrow{a} (A,a)) \\
&= s_{Ha} \circ Ha \circ r_{Ha} \\
&= s_{Ha} \circ r_{Ha} \circ s_{Ha} \circ r_{Ha} \\
&= id_{R_{Ha}} \\
&= id_{\bar{H}((A,a))} .
\end{aligned}$$

Now let

$$\begin{tikzcd}
(A,a) \arrow[r, "f"] \arrow[rr, bend right, "g \circ f"'] & (B,b) \arrow[r, "g"] & (C,c)
\end{tikzcd}$$

be given.

$$\begin{aligned}
\bar{H}(g) \circ \bar{H}(f) &= s_{Hc} \circ Hg \circ r_{Hb} \circ s_{Hb} \circ Hf \circ r_{Ha} \\
&= s_{Hc} \circ Hg \circ Hb \circ Hf \circ r_{Ha} \\
&= s_{Hc} \circ H(g \circ b \circ f) \circ r_{Ha} \\
&= s_{Hc} \circ H(g \circ f) \circ r_{Ha} \\
&= \bar{H}(g \circ f) .
\end{aligned}$$

Therefore $\bar{H}$ is a functor.

(d2) $\bar{H}$ preserves explicit splitting:

To show finally that $\bar{H}$ preserves explicit splitting we make use of the axiom of partial splitting in $\mathcal{B}$. For, let

$$\begin{tikzcd}
(A,a) \arrow[rr, "f"] \arrow[dr, "f"'] & & (A,a) \\
& (A,f) \arrow[ur, "f"'] &
\end{tikzcd}$$

be a splitting in $\mathcal{K}(A)$.

Taking its image under $\bar{H}$ we arrive at

$$\begin{tikzcd}
R_{Ha} \arrow[rr, "s_{Ha} \circ Hf \circ r_{Ha}"] \arrow[dr, "s_{Hf} \circ Hf \circ r_{Ha} = s_{Hf} \circ r_{Ha}"'] & & R_{Ha} \\
& R_{Hf} \arrow[ur, "s_{Ha} \circ Hf \circ r_{Hf} = s_{Ha} \circ r_{Hf}"'] &
\end{tikzcd}.$$

Now note that $Hf \subseteq Ha$ and that $Hf \upharpoonright Ha = s_{Ha} \circ Hf \circ r_{Ha}$.
By (APS) in $\mathcal{B}$ we have

$R_{Hf} = R_{Hf \upharpoonright Ha}$, $s_{Hf} = s_{Hf \upharpoonright Ha} \circ s_{Ha}$ and $r_{Hf} = r_{Ha} \circ r_{Hf \upharpoonright Ha}$.

$$\begin{tikzcd}
HA \arrow[rr, "Hf"] \arrow[d, "s_{Ha}"'] & & HA \\
R_{Ha} \arrow[rr, "Hf \upharpoonright Ha"] \arrow[dr, "s_{Hf \upharpoonright Ha}"'] & & R_{Ha} \arrow[u, "r_{Ha}"'] \\
& R_{Hf \upharpoonright Ha} \arrow[ur, "r_{Hf \upharpoonright Ha}"'] &
\end{tikzcd}.$$

It follows that $s_{Hf \upharpoonright Ha} = s_{Hf} \circ r_{Ha}$ and $r_{Hf \upharpoonright Ha} = s_{Ha} \circ r_{Hf}$, what had to be proved.

□

2.2.5. Remark. The original construction of Karoubi [1978], chapter I theorem 6.10., deals only with additive categories and embeds every category fully and faithfully in another one in which every idempotent has a kernel. To establish the connection with this theorem, note that p is idempotent if and only if (id-p) is idempotent and that ker(id-p) is the same as the equalizer of p and id, which is again the same as the section of the retract belonging to p.
By adding the structure of splitting to the category we were able

to define the Karoubi-construction as a left adjoint to the forgetful functor.

□

The Karoubi-construction will mainly be applied to a category with only one object, that is to say to a monoid. It is worth repeating the definition for this case.

2.2.6. Definition. Let $M = (X,*,u)$ be a monoid. Then the *Karoubi-envelope of* M (notation $K(M)$) is the Karoubi-functor applied to M considered as a category, that is

$$|K(M)| = \{a \in X \mid a * a = a\},$$

$$K(M)(a,b) = \{m \in X \mid b * m * a = m\},$$

$$\mathrm{id}_a = a \;,\; m \circ n = m * n.$$

□

2.2.7. Definition. Let C be a category. An object $U \in |C|$ is called *universal* if each object $A \in |C|$ is a retract of U, that is $A \lhd U$.

□

2.2.8. Lemma. Let $M = (X,*,u)$ be a monoid.

(i) $m \in K(M)(a,b) \iff b * m = m = m * a$.

(ii) $u \in |K(M)|$ is universal.

Proof:

(i) $\Rightarrow$: $b*m = b*b*m*a = b*m*a = m$.

Similarly $m * a = m$.

$\Leftarrow$: $b*m*a = m*a = m$.

(ii) Indeed $u \in |K(M)|$, since $u*u = u$.

For any $a \in |K(M)|$ we have (a,a): $a \lhd u$.

□

2.3. *Cartesian closed monoids.*

In this section we will transfer the cartesian closed structure of a categorical lambda algebra C to the monoid $C(U,U)$, thereby arriving at the notion of cartesian closed monoid. By proposition 2.2.2.(i) the objects $U \times U$ and U^U are both retracts of U. The construction will depend on the particular choice of these retracts.

2.3.1. Notation. Let C be a categorical lambda algebra. Consider retracts $(g_{U\times U}, f_{U\times U})\colon U\times U \lhd U$ and $(g_{U^U}, f_{U^U})\colon U^U \lhd U$. We will always assume that $g_{U^U} = G$ and $f_{U^U} = F$; otherwise we consider this new categorical lambda algebra. We say that these retracts are *canonical*, if moreover

$$g_{U\times U} = [\![\lambda z.zxy]\!]_{x,y} \circ [p_{UU}, q_{UU}],$$

$$f_{U\times U} = \langle [\![xK]\!]_x, [\![x(KI)]\!]_x \rangle \circ [id_U].$$

(This is exactly the retract defined in proposition 2.2.2.(i).)

The next definition gives the cartesian closed structure of the monoid $C(U,U)$ inherited from the cartesian closed structure of the category C.

2.3.2. Definition. Let C be a categorical lambda algebra. Let $X = C(U,U)$ and define binary operations $*, (\cdot,\cdot)$ on X, a unary operation $L(\cdot)$ on X and constants u,p,q,e in X by

$$\begin{aligned}
m * n &= m \circ n \qquad , \text{ for } m,n \in X,\\
u &= id_U\\
p &= p_{UU} \circ f_{U\times U}\\
q &= q_{UU} \circ f_{U\times U}\\
(m,n) &= g_{U\times U} \circ \langle m,n\rangle, \text{ for } m,n \in X,\\
e &= ev_{UU} \circ (f_{U^U} \times id_U) \circ f_{U\times U}\\
L(m) &= g_{U^U} \circ \Lambda(m \circ g_{U\times U}), \text{ for } m \in X.
\end{aligned}$$

Let $M(C) = (X,*,u,p,q,(\cdot,\cdot),e,L(\cdot))$ be the induced structure. □

The next lemma shows that this definition captures a notion of "cartesian-closedness" for a monoid.

2.3.3. Lemma. Let C be a categorical lambda algebra. Then in $M(C)$ for all $m,n,l \in |M(C)|$

(a) $p * (m,n) = m,\ q * (m,n) = n,$

(b) $(m,n) * l = (m * l, n * l),$

(c) $e * (L(m) * p, q) = m * (p,q),$

(d) $L(e * (m * p,q)) = L(e) * m$,

(e) $e * (p,q) = e$,

(f) $L(m * (p,q)) = L(m)$.

Proof:

(a) $$\begin{aligned} p * (m,n) &= p_{UU} \circ f_{U\times U} \circ g_{U\times U} \circ \langle m,n \rangle \\ &= p_{UU} \circ \langle m,n \rangle \\ &= m \ , \end{aligned}$$

similarly $q * (m,n) = n$.

(b) $$\begin{aligned} (m,n) * 1 &= g_{U\times U} \circ \langle m,n \rangle \circ 1 \\ &= g_{U\times U} \circ \langle m \circ 1, n \circ 1 \rangle \\ &= (m * 1, n * 1). \end{aligned}$$

(c) $$\begin{aligned} e * (L(m) * p,q) &= ev_{UU} \circ (f_{U^U} \times id_U) \circ f_{U\times U} \circ g_{U\times U} \circ \langle L(m) \circ p, q \rangle \\ &= ev_{UU} \circ ((f_{U^U} \circ g_{U^U} \circ \Lambda(m \circ g_{U\times U})) \times id_U) \circ f_{U\times U} \\ &= m \circ g_{U\times U} \circ f_{U\times U} \\ &= m * (p,q). \end{aligned}$$

(d) $$\begin{aligned} L(e * (m * p,q)) &= g_{U^U} \circ \Lambda(ev_{UU} \circ (f_{U^U} \times id_U) \circ f_{U\times U} \circ (m * p,q) \circ g_{U\times U}) \\ &= g_{U^U} \circ \Lambda(ev_{UU} \circ (f_{U^U} \times id_U) \circ \langle m \circ p_{UU}, q_{UU} \rangle) \\ &= g_{U^U} \circ \Lambda(ev_{UU} \circ ((f_{U^U} \circ m) \times id_U)) \\ &= g_{U^U} \circ f_{U^U} \circ m, \end{aligned}$$

$L(e) = g_{U^U} \circ \Lambda(ev_{UU} \circ (f_{U^U} \times id_U) \circ f_{U\times U} \circ g_{U\times U}) = g_{U^U} \circ f_{U^U}$.

Hence $L(e * (m * p,q)) = L(e) * m$.

(e) $$\begin{aligned} e * (p,q) &= ev_{UU} \circ (f_{U^U} \times id_U) \circ f_{U\times U} \circ g_{U\times U} \circ \langle p_{UU}, q_{UU} \rangle \circ f_{U\times U} \\ &= ev_{UU} \circ (f_{U^U} \times id_U) \circ f_{U\times U} \\ &= e. \end{aligned}$$

(f) $$\begin{aligned} L(m * (p,q)) &= g_{U^U} \circ \Lambda(m \circ g_{U\times U} \circ \langle p_{UU}, q_{UU} \rangle \circ f_{U\times U} \circ g_{U\times U}) \\ &= g_{U^U} \circ \Lambda(m \circ g_{U\times U}) \\ &= L(m). \end{aligned}$$

□

It is worth to collect these properties in a formal definition for an arbitrary monoid. We will reformulate properties (c) and (d) in such a way that they are equivalent to the old (c) and (d) under the assumption of (e) and (f), but that moreover the first four properties, (a), (b), (c) and (d), are sufficient to construct a categorical lambda algebra from a monoid with extra structure that satisfies them. The role of the peculiar properties (e) and (f) will be illustrated in proposition 2.3.8.

2.3.4. Definition.

(i) A *cartesian closed monoid* is a structure $\mathcal{M} = (X,*,u,p,q,(\cdot,\cdot),e,L(\cdot))$, where $u,p,q,e \in X$, $*,(\cdot,\cdot): X^2 \to X$ and $L(\cdot): X \to X$, such that $(X,*,u)$ is a monoid and for all $a,b,c \in X$

(a) $p * (a,b) = a$, $q * (a,b) = b$,

(b) $(a,b) * c = (a * c, b * c)$,

(c) $e * (L(a * (p,q)) * p, q) = a * (p,q)$,

(d) $L(e * (a * p, q)) = L(e * (p,q)) * a$.

(ii) A cartesian closed monoid $\mathcal{M}$ is *stable* if moreover

(e) $e * (p,q) = e$,

(f) $L(a * (p,q)) = L(a)$.

□

From now on we call $\mathcal{M}(\mathcal{C})$ the *cartesian closed monoid induced by* the categorical lambda algebra $\mathcal{C}$. $\mathcal{M}(\mathcal{C})$ is stable by lemma 2.3.3.
We will use the following abbreviations:

$$a \otimes b = (a * p, b * q),$$
$$\tilde{a} = a * (u \otimes u) = a * (p,q).$$

Then 2.3.4.(c),(d),(e),(f) can be reformulated as

(c) $e * (L(\tilde{a}) \otimes u) = \tilde{a}$,

(d) $L(e * (a \otimes u)) = L(\tilde{e}) * a$,

(e) $\tilde{e} = e$,

(f) $L(\tilde{a}) = L(a)$.

At first sight a cartesian closed monoid pays only attention to the product- and exponentiation -structure. It seems to forget about the terminal object. But this is treated implicitly as is seen from part (d) of the next lemma.

2.3.5. Lemma. Let $\mathcal{M}$ be a cartesian closed monoid. Then for all $a,b,c,d \in |\mathcal{M}|$

(a) $(a \otimes b) * (c \otimes d) = (a * c) \otimes (b * d)$,

(b) $L(\tilde{a}) = L(\tilde{e}) * L(\tilde{a})$,

(c) $L(a * (b \otimes u)) = L(\tilde{a}) * b$,

(d) $L(q) * a = L(q)$.

Proof:

(a) $$\begin{aligned}(a \otimes b) * (c \otimes d) &= (a * p, b * q) * (c * p, d * q)\\ &= (a * c * p, b * d * q), \text{ by 2.3.4.(a),(b)},\\ &= (a * c) \otimes (b * d).\end{aligned}$$

(b) $$\begin{aligned}L(\tilde{a}) &= L(e * (L(\tilde{a}) \otimes u)), \text{ by 2.3.4.(c)},\\ &= L(\tilde{e}) * L(\tilde{a}) \quad, \text{ by 2.3.4.(d)}.\end{aligned}$$

(c) $$\begin{aligned}L(a * (b \otimes u)) &= L(\tilde{a} * (b \otimes u)), \text{ by (a)},\\ &= L(e * ((L(\tilde{a}) * b) \otimes u)), \text{ by (a), 2.3.4.(c)},\\ &= L(\tilde{e}) * L(\tilde{a}) * b \quad, \text{ by 2.3.4.(d)},\\ &= L(\tilde{a}) * b \quad, \text{ by (b)}.\end{aligned}$$

(d) $$\begin{aligned}L(q) * a &= L(\tilde{q}) * a \quad, \text{ because } \tilde{q} = q,\\ &= L(q * (a \otimes u)), \text{ by (c)},\\ &= L(q).\end{aligned}$$

□

Now, given a cartesian closed monoid M, we want to define a cartesian closed category C with a reflexive object U. The ground for this construction has been laid in section 2.2.

2.3.6. <u>Definition</u>. Let $M = (X,*,u,p,q,(\cdot,\cdot),e,L(\cdot))$ be a structure with $(X,*,u)$ a monoid. Then $K(M)$, the *Karoubi-envelope of* M, is the category $K((X,*,u))$ together with the following structure:

(a) $T = L(q)$, with for any $a \in |K(M)|$ the map $!_a = L(q)$.

(b) For any $a,b \in |K(M)|$ let

$$a \times b = a \otimes b,$$
$$p_{ab} = a * p,$$
$$q_{ab} = b * q.$$

For any $c \in |K(M)|$, $m: c \to a$, $n: c \to b$, let $\langle m,n \rangle = (m,n)$.

(c) For any $a,b \in |K(M)|$ let

$$b^a = L(b * e * (u \otimes a)),$$
$$ev_{ab} = b * e * (u \otimes a).$$

For any $c \in |K(M)|$, $m: c\times a \to b$, let

$$\Lambda_{abc}(m) = L(m).$$

(d) Let $U = u$, $F = u^u = L(\tilde{e})$, $G = u^u = L(\tilde{e})$.

(e) The retracts $U\times U \lhd U$ and $U^U \lhd U$ are given by

$$g_{U\times U} = f_{U\times U} = u \otimes u = (p,q),$$
$$g_{U^U} = f_{U^U} = u^u = L(\tilde{e}).$$

□

2.3.7. Theorem. Let $M = (X,*,u,p,q,(\cdot,\cdot),e,L(\cdot))$ be a structure with $(X,*,u)$ a monoid. Then M is a cartesian closed monoid $\iff$ $K(M)$ is a categorical lambda algebra.

Proof: For convenience we will write here C for $K(M)$.

($\Rightarrow$) : We will show that the structure given in definition 2.3.6.(a)-(c) is cartesian closed and that 2.3.6.(d) defines a reflexive object in C.

(a) $T \in |C|$, since $L(q) * L(q) = L(q)$, by lemma 2.3.5.(d).

Now
$$\begin{aligned} m \in C(a,T) &\iff L(q) * m * a = m \\ &\iff L(q) = m, \text{ by lemma 2.3.5.(d).} \end{aligned}$$

Hence $C(a,T) = \{!_a\}$.

(b) $a\times b \in |C|$, since

$$\begin{aligned} (a\times b) * (a\times b) &= (a \otimes b) * (a \otimes b) \\ &= (a * a) \otimes (b * b), \text{ by 2.3.5.(a),} \\ &= a \otimes b \\ &= a\times b. \end{aligned}$$

$p_{ab} \in C(a\times b, a)$, since

$$\begin{aligned} a * p_{ab} * (a\times b) &= a * a * p * (a \otimes b) \\ &= a * a * p \\ &= p_{ab}. \end{aligned}$$

Similarly $q_{ab} \in C(a\times b, b)$.

Now let $m: c \to a$, $n: c \to b$.

$\langle m,n\rangle \in C(c, a\times b)$, since

$$\begin{aligned} (a\times b) * \langle m,n\rangle * c &= (a \otimes b) * (m,n) * c \\ &= (a * m * c, b * n * c) \\ &= (m,n) \\ &= \langle m,n\rangle. \end{aligned}$$

For $1 \in \mathcal{C}(c, a\times b)$ we finally have

$$\begin{aligned} p_{ab} \circ 1 = m \wedge q_{ab} \circ 1 = n &\iff (p_{ab}, q_{ab}) * 1 = (m,n) \\ &\iff (a \otimes b) * 1 = (m,n) \\ &\iff 1 = \langle m,n \rangle . \end{aligned}$$

(c) First note the fact that $\widetilde{b * e * (u \otimes a)} = b * e * (u \otimes a)$.

$b^a \in |\mathcal{C}|$, since

$$\begin{aligned} b^a * b^a &= L(b * e * (u \otimes a)) * b^a \\ &= L(b * e * (u \otimes a) * (b^a \otimes u)), \text{ by 2.3.5.(c)}, \\ &= L(b * e * (b^a \otimes u) * (u \otimes a)) \\ &= L(b * (b * e * (u \otimes a)) * (u \otimes a)), \text{ by 2.3.4.(c)}, \\ &= L(b * e * (u \otimes a)) \\ &= b^a . \end{aligned}$$

$ev_{ab} \in \mathcal{C}(b^a \times a, b)$, since

$$\begin{aligned} b * ev_{ab} * (b^a \times a) &= b * b * e * (u \otimes a) * (b^a \otimes a) \\ &= b * e * (b^a \otimes u) * (u \otimes a) \\ &= b * (b * e * (u \otimes a)) * (u \otimes a) \\ &= b * e * u \otimes a \\ &= ev_{ab} . \end{aligned}$$

Now let $m \in \mathcal{C}(c \times a, b)$. Then $\Lambda(m) \in \mathcal{C}(c, b^a)$, since

$$\begin{aligned} b^a * \Lambda(m) * c &= L(b * e * (u \otimes a)) * L(m) * c \\ &= L(b * e * (u \otimes a) * ((L(m) * c) \otimes u)), \text{ by 2.3.5.(c)}, \\ &= L(b * e * (L(m) \otimes u) * (c \otimes a)) \\ &= L(b * m * (c \otimes a)), \text{ since } m = \widetilde{m}, \\ &= \Lambda(m) . \end{aligned}$$

For $n \in \mathcal{C}(c, b^a)$ we finally have

$$\begin{aligned} ev_{ab} \circ (n \times id_a) = m &\iff b * e * (n \otimes a) = m \\ &\iff L(b * e * (n \otimes a)) = L(m) \\ &\iff L(b * e * (u \otimes a) * (n \otimes u)) = L(m) \\ &\iff b^a * n = L(m) \\ &\iff n = \Lambda(m) . \end{aligned}$$

(The second equivalence holds because of 2.3.4.(c) and the fact that $\widetilde{m} = m$.)

(d) $U \in |\mathcal{C}|$ since $u * u = u$.

Then $$\begin{aligned} U^U = u^u &= L(u * e * (u \otimes u)) \\ &= L(\widetilde{e}) . \end{aligned}$$

Hence $(G,F) : U^U \lhd U$.

We conclude that $\mathcal{K}(M)$ is a categorical lambda algebra.

($\Leftarrow$): We have in $\mathcal{K}(M)$

$$p_{uu} = u * p = p \quad \text{and} \quad q_{uu} = u * q = q .$$

Furthermore, if $a,b \in |M|$, then $a,b \in \mathcal{K}(M)(u,u)$ and hence

$$\langle a,b \rangle = (a,b) \in \mathcal{K}(M)(u,u\times u).$$

Now 2.3.4.(a),(b) follow easily from the universal property of products in $\mathcal{K}(M)$

$$\begin{tikzcd}
 & & u \\
u \arrow[urr, bend left, "a * c"] \arrow[r, "c"'] \arrow[drr, bend right, "b * c"'] & u \arrow[ur, "a"] \arrow[r, "{(a,b)}"] \arrow[dr, "b"'] & u\times u \arrow[u, "p"'] \arrow[d, "q"] \\
 & & u
\end{tikzcd} \quad .$$

Also in $\mathcal{K}(M)$

$$u^u = L(\tilde e) \quad \text{and} \quad ev_{uu} = \tilde e :$$

$$\begin{tikzcd}
L(\tilde e)\times u \arrow[r, "\tilde e"] & u \\
u\times u \arrow[u, "\Lambda(\tilde a)\times id_u"] \arrow[ur, "\tilde a"'] &
\end{tikzcd} \qquad \Lambda(\tilde a) = L(\tilde a).$$

By the universal property of exponentiation in $\mathcal{K}(M)$ we conclude

$$\tilde e * (L(\tilde a) \otimes u) = \tilde a \quad \text{for all } a \in |M|$$

and

$$L(\tilde e * ((L(\tilde e) * b) \otimes u)) = L(\tilde e) * b \quad \text{for all } b \in |M| .$$

From the first equation it follows that $\tilde e * (L(\tilde e) \otimes u) = \tilde e$. Hence the second equation reduces to

$$L(\tilde e * (b \otimes u)) = L(\tilde e) * b .$$

Now we can prove 2.3.4.(c),(d) by observing that for all c we have

$$\tilde{e} * (c \otimes u) = e * (c \otimes u).$$

□

2.3.8. Proposition. Let $M = (X, *, u, p, q, (\cdot,\cdot), e, L(\cdot))$ be a cartesian closed monoid and let $\tilde{M} = M(K(M))$.

(i) $\tilde{M}$ is a stable cartesian closed monoid.

(ii) $\tilde{M} = (X, *, u, p, q, (\cdot,\cdot), \tilde{e}, \tilde{L}(\cdot))$ where $\tilde{L}(a) = L(\tilde{a})$.

(iii) $\tilde{M} = M \iff M$ is stable.

(iv) $K(\tilde{M}) = K(M)$.

Proof:

(i) This follows from lemma 2.3.3.

(ii) $|\tilde{M}| = K(M)(u,u) = \{a \in |M| \mid u * a * u = a\} = |M|$.

$a * b$ (in $\tilde{M}$) $= a \circ b$ (in $K(M)$) $= a * b$ (in M).

With a similar interpretation we also have

$$u = id_u = u,$$

$$\begin{aligned} p = p_{uu} \circ f_{u\times u} &= p * (u \otimes u) \\ &= p, \end{aligned}$$

$$\begin{aligned} q = q_{uu} \circ f_{u\times u} &= q * (u \otimes u) \\ &= q, \end{aligned}$$

$$\begin{aligned} (a,b) = g_{u\times u} \circ \langle a,b \rangle &= (u \otimes u) * (a,b) \\ &= (a,b), \end{aligned}$$

$$\begin{aligned} e = ev_{uu} \circ (f_{u^u} \times id_u) \circ f_{u\times u} &= e * (u \otimes u) * (L(\tilde{e}) \otimes u) * (u \otimes u) \\ &= e * (L(\tilde{e}) \otimes u) \\ &= \tilde{e}\ , \text{ by 2.3.4.(c)}, \end{aligned}$$

$$\begin{aligned} L(a) = g_{u^u} \circ \Lambda(a \circ g_{u\times u}) &= L(\tilde{e}) * L(a * u \otimes u) \\ &= L(\tilde{e}) * L(\tilde{a}) \\ &= L(\tilde{a}), \text{ by 2.3.5.(b)}. \end{aligned}$$

(iii) $\tilde{M} = M \iff \tilde{e} = e$ and $\tilde{L} = L$
$\iff \tilde{e} = e$ and $L(\tilde{a}) = L(a)$ for all $a \in X$
$\iff M$ is stable.

(iv) It makes no difference in definition 2.3.6. whether we take e and L or $\tilde{e}$ and $\tilde{L}$ as one easily checks. □

We refer the reader to Yokouchi [1983] for an analysis of the conditions under which $K(M(C)) \cong C$ holds for a categorical lambda algebra C.

The next interesting question is to analyse under which conditions M is such that the retracts in $K(M)$, as defined in 2.3.6.(e), are canonical. This analysis was given by Adachi [1983] and Yokouchi [1983]. These conditions are given in the next definition.

2.3.9. Definition. A cartesian closed monoid M is *canonical* if

(a) $p = e * \langle u, L(L(q * p))\rangle$,

(b) $q = e * \langle u, L(L(q))\rangle$,

(c) $u \otimes u = L(e * \langle e * \langle q, p * p\rangle, q * p\rangle)$. □

2.3.10. Proposition.

(i) Let C be a categorical lambda algebra. Then

C has canonical retracts $\Longleftrightarrow$ $M(C)$ is canonical.

(ii) Let M be a cartesian closed monoid. Then

$K(M)$ has canonical retracts $\Longleftrightarrow$ M is canonical.

Proof:

(i) Easy (but tedious) calculations. The amount of calculation can be considerably reduced by using lemma 2.4.5. in the next section.

(ii) $K(M)$ has canonical retracts $\Longleftrightarrow$ $\widetilde{M}$ is canonical, by (i),

$\Longleftrightarrow$ M is canonical. □

In the remainder of this text we will mostly use cartesian closed monoids that are stable and canonical. We will state the convention that the construction of $M(C)$ out of C will make use of the canonical retracts.

2.3.11. Definition.

(i) A *lambda monoid* is a canonical, stable cartesian closed monoid.

(ii) If C is a categorical lambda algebra (with canonical retracts) then $M(C)$ is called the *induced* lambda monoid. □

2.4. *Lambda algebras versus lambda monoids.*

It turns out that the lambda monoid $M(C)$ induced by the categorical lambda algebra C is determined completely in terms of the lambda algebra $\mathfrak{M}(C)$ generated by C. Using the Karoubi-construction we will then proceed to show that the other way around also the lambda monoid $M(C)$ determines the lambda algebra $\mathfrak{M}(C)$.

Since both constructions are in fact independent of C this will show that *any* lambda algebra $\mathfrak{M}$ is the lambda algebra generated by some categorical lambda algebra, viz. $K(\mathfrak{M})$, the Karoubi-envelope of the cartesian closed monoid constructed from $\mathfrak{M}$.

Defining $M(C)$ in terms of $\mathfrak{M}(C)$ depends on the existence of the following retract in C.

2.4.1. Definition. Let C be a categorical lambda algebra.

(i) Define α: $C(U,U) \to C(T,U)$ by

$$\alpha(m) = G \circ \Lambda(m \circ q_{TU}).$$

Intuitively, $\alpha(m) = [\![\text{"}\lambda x.m(x)\text{"}]\!]$.

(ii) Define β: $C(T,U) \to C(U,U)$ by

$$\beta(a) = ev \circ \langle F \circ a \circ !_U, id_U \rangle.$$

In fact, $\beta(a) = [\![\underline{a}x]\!]_x \circ [id]$.

□

2.4.2. Lemma. (α,β): $C(U,U) \lhd C(T,U)$ is a retract.

Proof:
$$\begin{aligned}\beta(\alpha(m)) &= ev \circ \langle F \circ G \circ \Lambda(m \circ q_{TU}) \circ !_U, id_U \rangle \\ &= m \circ q_{TU} \circ \langle !_U, id_U \rangle \\ &= m.\end{aligned}$$

□

Because of the injective mapping α we have that

$$|M(C)| \cong (\alpha |M(C)|) \subseteq |\mathfrak{M}(C)|.$$

In fact the whole structure of $M(C)$ is definable inside $\mathfrak{M}(C)$.

2.4.3. Proposition. Let C be a categorical lambda algebra. Then for $a \in |\mathfrak{M}(C)|, m, n \in |M(C)|$.

(i) $a \in \text{range}(\alpha) \iff a = [\![\lambda x.\underline{a}x]\!]$.

(ii) $\alpha(m * n) = [\![\lambda x.\underline{\alpha(m)}\,\underline{(\alpha(n)}x)]\!]$.

(iii) $\alpha(u) = [\![\lambda x.x]\!]$.

(iv) $\alpha(p) = \llbracket \lambda x.xK \rrbracket$.

(v) $\alpha(q) = \llbracket \lambda x.x(KI) \rrbracket$.

(vi) $\alpha((m,n)) = \llbracket \lambda x.[\underline{\alpha(m)}x,\underline{\alpha(n)}x] \rrbracket$. (Here $[M,N] = \lambda z.zMN$.)

(vii) $\alpha(e) = \llbracket \lambda x.xK(x(KI)) \rrbracket$.

(viii) $\alpha(L(m)) = \llbracket \lambda xy.\underline{\alpha(m)}[x,y] \rrbracket$.

Proof: Routine calculations, e.g.

(iv) $$\begin{aligned} \llbracket \lambda x.xK \rrbracket &= G \circ \Lambda(\llbracket xK \rrbracket_x) \\ &= G \circ \Lambda(p_{UU} \circ f_{U\times U} \circ q_{TU}), \text{ see 2.3.1.,} \\ &= \alpha(p_{UU} \circ f_{U\times U}), \\ &= \alpha(p) \quad \text{, by 2.3.2.} \end{aligned}$$

□

Inspired by this proposition we define a structure $M(\mathfrak{M})$ for an arbitrary lambda algebra $\mathfrak{M}$.

2.4.4. Definition. Let $\mathfrak{M}$ be a lambda algebra. Define

$$\begin{aligned} |M(\mathfrak{M})| &= \{a \in |\mathfrak{M}| \mid a = \llbracket \lambda x.\underline{a}x \rrbracket\}, \\ u &= \llbracket \lambda x.x \rrbracket, \\ p &= \llbracket \lambda x.xK \rrbracket, \\ q &= \llbracket \lambda x.x(KI) \rrbracket, \\ e &= \llbracket \lambda x.xK(x(KI)) \rrbracket; \end{aligned}$$

and for all $a,b \in |M(\mathfrak{M})|$

$$\begin{aligned} a * b &= \llbracket \lambda x.\underline{a}(\underline{b}x) \rrbracket, \\ (a,b) &= \llbracket \lambda x.[\underline{a}x,\underline{b}x] \rrbracket, \\ L(a) &= \llbracket \lambda xy.\underline{a}[x,y] \rrbracket. \end{aligned}$$

This defines a structure $M(\mathfrak{M})$.

□

2.4.5. Lemma. Let $\mathfrak{M}$ be a lambda algebra. Then $M(\mathfrak{M})$ is a lambda monoid.

Proof: Routine calculations, e.g.

$$\begin{aligned} p * (a,b) &= \lambda x.p((a,b)x) \\ &= \lambda x.(a,b)xK \\ &= \lambda x.ax \\ &= a, \end{aligned}$$

which proves part of property 2.3.4.(a).

□

Note that, since $M(C) \cong M(\mathfrak{M}(C))$ (by proposition 2.4.3. and definition

2.4.4.), lemma 2.4.5. gives an easier proof for proposition 2.3.10., because computations in $M(C)$ can now make use of lambda calculus equations, which are valid in $\mathfrak{M}(C)$.

In general we have the following situation

$$\begin{tikzcd} |M(C)| \arrow[r, shift left, "\alpha"] \arrow[r, phantom, "\simeq"] & |M(\mathfrak{M}(C))| \arrow[l, shift left, "\beta"] \subseteq |\mathfrak{M}(C)| . \end{tikzcd}$$

In the case that $C = K(M)$, where M is a lambda monoid, this reduces to

$$\begin{tikzcd} |M| \arrow[r, shift left, "\alpha"] \arrow[r, phantom, "\simeq"] & |M(\mathfrak{M}(K(M)))| \arrow[l, shift left, "\beta"] \subseteq |\mathfrak{M}(K(M))| , \end{tikzcd}$$

where
$$\begin{aligned} \alpha(a) &= G \circ \Lambda(a \circ q_{TU}) \\ &= L(e) * L(a * q) \\ &= L(a * q) . \end{aligned}$$

Finally, when $M = M(\mathfrak{M})$, then this reduces to
$$\begin{aligned} \alpha(a) &= L(a * q) \\ &= \lambda xy.a(q[x,y]) \\ &= \lambda xy.a([x,y](KI)) \\ &= \lambda xy.ay \\ &= ka, \text{ for } a \in |M| = \{a \in |\mathfrak{M}| \mid a = \lambda x.ax\}. \end{aligned}$$

Our next purpose is to show that the mapping α: $a \mapsto ka$ with domain $|M(\mathfrak{M})|$ can be extended to the whole of $|\mathfrak{M}|$, resulting in an isomorphism between $\mathfrak{M}$ and $\mathfrak{M}(K(M(\mathfrak{M})))$. In order to accomplish this we have to know how the interpretation in $K(M)$ looks like.

2.4.6. Definition. Let M be a cartesian closed monoid. For any $M \in \Lambda(|M|)$ and $|\Delta| \supseteq FV(M)$ we define $|\Delta : M| \in |M|$ by induction on M as follows

$$\begin{aligned} |\Delta : \underline{a}| &= a * L(q), \\ |\Delta, y : x| &= |\Delta : x| * p \text{ ,if } y \neq x, \\ & \quad q \text{ ,if } y = x, \\ |\Delta : MN| &= e * (|\Delta : M|, |\Delta : N|), \\ |\Delta : \lambda x.M| &= L(|\Delta, x : M|), \text{where } x \notin \Delta. \end{aligned}$$

□

2.4.7. Proposition. Let M be a cartesian closed monoid. Then for the interpretation $[\![\cdot]\!]$ in $\mathcal{K}(M)$ we have

$$[\![M]\!]_\Delta = |\Delta : M|, \text{ for all } M \in \Lambda(\mathfrak{M}(\mathcal{K}(M))).$$

Proof: Let us first note the following property:

$$(*) \quad \pi_x^{\Delta,y} = \begin{cases} \pi_x^{\Delta} * p & ,\text{if } y \neq x, \\ q & ,\text{if } y = x. \end{cases}$$

$$(*) \quad \text{follows from } \; p_{X\Delta,U} = (X\Delta) * p,$$

$$q_{X\Delta,U} = u * q = q,$$

$$\pi_x^{\Delta} * (X\Delta) = \pi_x^{\Delta}.$$

Now we prove the result by induction on M.

$$\begin{aligned} [\![\underline{a}]\!]_\Delta &= a \circ \Pi^{\Delta}_{\langle\,\rangle} \\ &= a * L(q) \\ &= |\Delta : \underline{a}|. \end{aligned}$$

$[\![x]\!]_\Delta = |\Delta : x|$ by induction on Δ, using $(*)$.

$$\begin{aligned} [\![MN]\!]_\Delta &= ev \circ \langle F \circ [\![M]\!]_\Delta, [\![N]\!]_\Delta \rangle \\ &= e * (u \otimes u) * (L(\tilde{e}) * [\![M]\!]_\Delta, [\![N]\!]_\Delta) \\ &= \tilde{e} * ([\![M]\!]_\Delta, [\![N]\!]_\Delta) \\ &= |\Delta : MN| \end{aligned}$$

$$\begin{aligned} [\![\lambda x{\cdot}M]\!]_\Delta &= G \circ \Lambda([\![M]\!]_{\Delta,x}) \\ &= L(\tilde{e}) * L([\![M]\!]_{\Delta,x}) \\ &= L([\![M]\!]_{\Delta,x}), \text{ by 2.3.5.(b), since } [\![M]\!]_{\Delta,x} = [\![\tilde{M}]\!]_{\Delta,x}, \\ &= L(|\Delta,x : M|), \text{ by induction hypothesis,} \\ &= |\Delta : \lambda x{\cdot}M|. \end{aligned}$$

□

2.4.8. Definition. Let $A,B \in \Lambda(C)$. Then A^nB and $AB^{\sim n}$ are defined by induction on $n \in \omega$ as

$$A^0B = B\,, \quad A^{n+1}B = A(A^nB)\,,$$
$$AB^{\sim 0} = A\,, \quad AB^{\sim n+1} = (AB^{\sim n})B\,.$$

□

2.4.9. Proposition. Let $\mathfrak{M}$ be a lambda algebra and $\mathcal{M}(\mathfrak{M})$ be the corresponding lambda monoid. If $M \in \Lambda(|\mathcal{M}(\mathfrak{M})|)$ and $\Delta = x_1,\ldots,x_n$ with $|\Delta| \supseteq FV(M)$, then

$$|\Delta : M| = \lambda z.M[\vec{\underline{a}} := \vec{\underline{a}I},\ldots,x_i := zK^{\sim n-i}(KI),\ldots].$$

Proof: By induction on $M \in \Lambda(|\mathcal{M}(\mathfrak{M})|)$.

$$\begin{aligned} |\Delta : \underline{a}| &= a * L(q) \\ &= \lambda z.\underline{a}I\,. \end{aligned}$$

Now let $\Gamma = x_1,\ldots,x_{n-1}$. Then

$$\begin{aligned} |\Gamma,x_n : x_n| &= q \\ &= \lambda z.z(KI) \\ &= \lambda z.x_n[x_n := zK^{\sim n-n}(KI)]\,, \end{aligned}$$

and for $i<n$, using induction on n,

$$\begin{aligned} |\Gamma,x_n : x_i| &= |\Gamma : x_i| * p \\ &= \lambda z.(\lambda w.x_i[x_i := wK^{\sim n-1-i}(KI)])(zK), \text{ by IH}, \\ &= \lambda z.x_i[x_i := zK^{\sim n-i}(KI)]\,. \end{aligned}$$

$$\begin{aligned} |\Delta : MN| &= e*(|\Delta : M|, |\Delta : N|) \\ &= \lambda z.e[\,|\Delta : M|z, |\Delta : N|z] \\ &= \lambda z.\,|\Delta : M|z\,(|\Delta : N|z) \\ &= \lambda z.M[\vec{\underline{a}} := \vec{\underline{a}I},\ldots]\ N[\vec{\underline{a}} := \vec{\underline{a}I},\ldots], \text{ by IH}, \\ &= \lambda z.(MN)[\vec{\underline{a}} := \vec{\underline{a}I},\ldots]\,. \end{aligned}$$

$$\begin{aligned} |\Delta : \lambda x.M| &= L(\,|\Delta,x : M|) \\ &= \lambda zy.\,|\Delta,x : M|[z,y] \\ &= \lambda zy.M[\vec{\underline{a}} := \vec{\underline{a}I},\ldots,x_i := [z,y]K^{\sim n+1-i}(KI),\ldots \\ &\qquad \ldots,x := [z,y](KI)] \end{aligned}$$

$$= \lambda zy.M[\underline{\vec{a}} := \underline{\vec{a}}I,\ldots,x_i := zK^{\sim n-i}(KI),\ldots,x := y]$$
$$= \lambda z.(\lambda x.M)[\underline{\vec{a}} := \underline{\vec{a}}I,\ldots,x_i := zK^{\sim n-i}(KI),\ldots].$$
□

From now on let $\mathfrak{M}(\mathcal{M})$ denote $\mathfrak{M}(\mathcal{K}(\mathcal{M}))$. Then combining propositions 2.4.7. and 2.4.9. we may show the main result of this chapter, which says that any lambda algebra $\mathfrak{M}$ can be represented as the lambda algebra induced by a categorical lambda algebra, viz. $\mathcal{K}(\mathcal{M}(\mathfrak{M}))$.

2.4.10. Theorem. Let $\mathfrak{M}$ be any lambda algebra. Then $\alpha(a) = ka$ defines an isomorphism from $\mathfrak{M}$ onto $\mathfrak{M}(\mathcal{M}(\mathfrak{M}))$.

Proof: Let us put $\mathfrak{M}' = \mathfrak{M}(\mathcal{M}(\mathfrak{M}))$. Then we know

$$\begin{aligned} b \in |\mathfrak{M}'| &\iff b: L(q) \to u \quad \text{in } \mathcal{K}(\mathcal{M}(\mathfrak{M})) \\ &\iff b * L(q) = b \quad \text{in } \mathcal{M}(\mathfrak{M}) \\ &\iff \lambda z.bI = b \quad \text{in } \mathfrak{M} \\ &\iff \exists a \in |\mathfrak{M}| \ (b = ka). \end{aligned}$$

This shows that α is surjective.
Clearly α is injective, since $ka = ka' \Rightarrow kax = ka'x \Rightarrow a = a'$.
Hence α is a bijective mapping.

It remains to show, by lemma 1.6.3., that α preserves interpretation, that is

$$\alpha([\![M]\!]^{\mathfrak{M}}) = [\![M[\underline{\vec{a}} := \underline{\overrightarrow{\alpha(a)}}]]\!]^{\mathfrak{M}'}$$

for all $M \in \Lambda^0(\mathfrak{M})$.

Indeed

$$\begin{aligned} [\![M[\underline{\vec{a}} := \underline{\overrightarrow{\alpha(a)}}]]\!]^{\mathfrak{M}'} &= |\langle\,\rangle : M[\underline{\vec{a}} := \underline{\overrightarrow{ka}}]|, \quad \text{by 2.4.7.,} \\ &= \lambda z.M[\underline{\vec{a}} := \underline{\overrightarrow{ka\,I}}] \ , \quad \text{by 2.4.9.,} \\ &= k[\![M[\underline{\vec{a}} := \underline{\vec{a}}]]\!]^{\mathfrak{M}} \\ &= \alpha([\![M]\!]^{\mathfrak{M}}). \end{aligned}$$

□

As a corollary to this theorem and proposition 2.3.8. we may state that the category of lambda algebras is equivalent to the category of lambda monoids. For a further functorial analysis of the constructions in this chapter we refer the reader to Adachi [1983] and Yokouchi [1983].

2.4.11. Summary. Let $\mathcal{C}$ be a categorical lambda algebra, $\mathfrak{M}$ a lambda algebra and $\mathcal{M}$ a lambda monoid. Then we have the following constructions

$$\begin{tikzcd}
\mathcal{C} \arrow[rr, "\mathfrak{M}\ (2.1.4.)"] \arrow[dr, dashed, bend right, "\mathcal{M}(2.3.2.)"'] & & \mathfrak{M} \arrow[dl, "\mathcal{M}(2.4.4.)"] \\
& \mathcal{M} \arrow[ul, "\mathcal{K}(2.2.6.)"] &
\end{tikzcd}$$
.

We write $\mathfrak{M}(\mathcal{M})$ for $\mathfrak{M}(\mathcal{K}(\mathcal{M}))$
and $\mathcal{K}(\mathfrak{M})$ for $\mathcal{K}(\mathcal{M}(\mathfrak{M}))$.

We showed the following facts

(1) $\mathcal{M}(\mathcal{K}(\mathcal{M})) = \mathcal{M}$ (proposition 2.3.8.)

(2) $\mathcal{M}(\mathcal{C}) \cong \mathcal{M}(\mathfrak{M}(\mathcal{C}))$ (remark after lemma 2.4.5.)

(3) $\mathcal{M}(\mathfrak{M}(\mathcal{M})) \cong \mathcal{M}$ (follows from (1) and (2))

(4) $\mathfrak{M}(\mathcal{M}(\mathfrak{M})) \cong \mathfrak{M}$ (theorem 2.4.10.).

We mention here again that in general $\mathcal{K}(\mathcal{M}(\mathcal{C})) \ncong \mathcal{C}$, because $\mathcal{C}$ may contain many objects that are not a retract of the reflexive object U. On the other hand, if we restrict attention to $\mathcal{C}\upharpoonright$ U, the subcartesian closed category of $\mathcal{C}$, generated by U, and if we fix retracts for every object in $\mathcal{C}\upharpoonright$ U then we may show $\mathcal{C}\upharpoonright \mathrm{U} \cong \mathcal{K}(\mathcal{M}(\mathcal{C}))$. For details we refer to Yokouchi [1983].

Since we are mostly interested in the connection between $\mathfrak{M}$ and $\mathcal{K}(\mathfrak{M})$ we work with the following notation.

For $a,b \in |\mathfrak{M}|$, $a \circ b = \lambda x.a(bx)$.

$|\mathcal{K}(\mathfrak{M})| = \{a \in |\mathfrak{M}| \mid a \circ a = a\}$,

$\mathcal{K}(\mathfrak{M})(a,b) = \{f \in |\mathfrak{M}| \mid b \circ f \circ a = f\}$.

$T = KI$ is the terminal object in $\mathcal{K}(\mathfrak{M})$, $!_a = KI$,

$a \times b = \lambda x.[a(xK), b(x(KI))]$, $p_{ab} = \lambda x.a(xK)$, $q_{ab} = \lambda x.b(x(KI))$,

$b^a = \lambda x.b \circ x \circ a$, $ev_{ab} = \lambda x.b(xK(a(x(KI))))$.

$\langle f,g \rangle = \lambda x.[fx,gx]$, $\Lambda(f) = \lambda xy.f[x,y]$. □

2.5. *Lambda models, concrete models and extensionality.*

We remind the reader of definition 2.1.12. We just say that a category has enough points, if it has enough points at every object.

2.5.1. Lemma. Let $\mathfrak{M}$ be a lambda algebra. Then

(i) $\mathfrak{M}$ is a lambda model $\Longleftrightarrow$ $K(\mathfrak{M})$ has enough points.

(ii) $\mathfrak{M} \models 1 = I$ $\Longleftrightarrow$ $K(\mathfrak{M}) \models G \circ F = id_U$.

Proof:

(i) $\Rightarrow$: Let $a,b \in |K(\mathfrak{M})|$, $f,g \in K(\mathfrak{M})(a,b)$.
Suppose $\forall x \in K(\mathfrak{M})(T,a)(f \circ x = g \circ x)$.
Let $d \in |\mathfrak{M}|$. Then $k(ad) \in K(\mathfrak{M})(T,a)$.
Hence $f \circ (k(ad)) = g \circ (k(ad))$,
so $k(f(ad)) = k(g(ad))$,
so $(f \circ a)d = (g \circ a)d$.
But $f \circ a = f$, $g \circ a = g$ since f,g: $a \to b$.
Therefore $\forall d \in |\mathfrak{M}|\ (fd = gd)$.
By weak extensionality $\lambda x.fx = \lambda x.gx$. But
$f = b \circ f = \lambda x.(b \circ f)x = \lambda x.fx$ and similarly $g = \lambda x.gx$.
We conclude $f = g$. Hence $K(\mathfrak{M})$ has enough points.

$\Leftarrow$: $\mathfrak{M} \cong \mathfrak{M}(K(\mathfrak{M}))$, which is weakly extensional by proposition 2.1.13.(i).

(ii) $\mathfrak{M} \models 1 = I$ $\Longleftrightarrow$ $\mathfrak{M}(K(\mathfrak{M})) \models 1 = I$

$\Longleftrightarrow$ $K(\mathfrak{M}) \models G \circ F = id_U$ by proposition 2.1.13.(ii).

□

2.5.2. Corollary. Let $\mathfrak{M}$ be a lambda algebra. Then
$\mathfrak{M}$ is an extensional lambda model $\Longleftrightarrow$ $K(\mathfrak{M})$ has enough points and G: $U^U \cong U$ with $F = G^{-1}$.

Proof: By proposition 1.5.3. and the above lemma.

□

In many cases the categorical lambda algebra C is such that the objects of C are sets with a certain structure and the arrows of C are just ordinary functions, satisfying some property. For instance the category CPO has as objects complete partial orders and as arrows continuous functions with respect to the Scott topology.

If one works in such a category the generated lambda algebra is in fact a lambda model and the definition of interpretation can be simplified.

2.5.3. Definition. Let C be a categorical lambda algebra.

C is *strictly concrete* (via H) if $H: C \to \mathrm{Set}$ is a functor such that for all $A,B \in |C|$

(a) H is faithful,

(b) H is full on $C(T,A)$,

(c) HT is a singleton set, say $HT = \{*\}$,

$H(A\times B) = HA\times HB$,

$H(p_{AB}) = p_{HA,HB}$, $H(q_{AB}) = q_{HA,HB}$,

(d) $H(B^A) \subseteq HB^{HA}$,

$H(ev_{AB}) = ev_{HA,HB} \restriction H(B^A) \times H(A)$.

□

2.5.4. Remark. The fact that H is full and faithful on $C(T,A)$ implies

$HA \cong \mathrm{Set}(\{*\},HA) = \mathrm{Set}(HT,HA) \cong C(T,A)$,

hence $H \cong C(T,-)$.

On the other hand we have (for an arbitrary categorical lambda algebra C)

(a) $C(T,-)$ is faithful $\Longleftrightarrow$ C has enough points,

(b) $C(T,-)$ is always full on $C(T,A)$,

(c) $C(T,T) = \{id_T\}$,

$C(T,A\times B) \cong C(T,A) \times C(T,B)$,

(d) C has enough points $\Longleftrightarrow$ $(f \mapsto (a \mapsto ev_{AB} \circ \langle f,a\rangle))$:

$C(T,B^A) \to C(T,B)^{C(T,A)}$ is an injective mapping.

Hence, for C to be strictly concrete, the important condition is that C has enough points. In definition 2.5.3. we demand that H commutes with products and exponentation exactly (and not only up to isomorphism) for easiness of interpretation.

□

2.5.5. Definition. Let C be a strictly concrete categorical lambda algebra via H. Define

$X = HU$,

$$\begin{aligned}
a\cdot b &= \mathrm{HF}(a)(b) \quad \text{for } a,b \in X,\\
[\![\cdot]\!] &: \Lambda^0(|X|) \to X \quad \text{by induction as}\\
[\![\underline{a}]\!] &= a \qquad\qquad , \text{ for } a \in X,\\
[\![MN]\!] &= [\![M]\!]\cdot[\![N]\!]\\
[\![\lambda x.M]\!] &= \mathrm{HG}(d \mapsto [\![M[x := \underline{d}]]\!]).
\end{aligned}$$

Let $\mathfrak{M}(C,H) = (X,\cdot,[\![\cdot]\!])$ be the *concrete model* induced by C,H.

□

2.5.6. Theorem. Let C be a strictly concrete categorical lambda algebra via H. Then $\mathfrak{M}(C,H)$ is a lambda model isomorphic to $\mathfrak{M}(C)$.

Proof: Let $\mathfrak{M} = \mathfrak{M}(C)$, $\mathfrak{N} = \mathfrak{M}(C,H)$. We show that $\varphi\colon |\mathfrak{M}| \to |\mathfrak{N}|$, defined by $\varphi(a) = Ha(*)$, is an isomorphism between $\mathfrak{M}$ and $\mathfrak{N}$.
φ is bijective, since H is full and faithful on $C(T,U)$.

Let us note the following properties of H:

$H(\langle f,g\rangle) = \langle Hf,Hg\rangle$, for

$$\begin{aligned}
Hf &= H(p \circ \langle f,g\rangle)\\
&= Hp \circ H(\langle f,g\rangle)\\
&= p \circ H(\langle f,g\rangle)\\
&\qquad \text{and similarly}\\
Hg &= q \circ H(\langle f,g\rangle).
\end{aligned}$$

$H(f\times g) = Hf\times Hg$, for

$$\begin{aligned}
H(f\times g) &= H(\langle f \circ p, g \circ q\rangle)\\
&= \langle Hf \circ Hp, Hg \circ Hq\rangle\\
&= \langle Hf \circ p, Hg \circ q\rangle\\
&= Hf\times Hg.
\end{aligned}$$

If $j\colon H(B^A) \xrightarrow{\subseteq} HB^{HA}$, then $j \circ H(\Lambda f) = \Lambda(Hf)$, for

$$\begin{aligned}
ev \circ (j \circ H(\Lambda f)) \times id &= ev \circ j\times id \circ H(\Lambda f)\times id\\
&= ev \restriction H(B^A)\times HA \circ H(\Lambda f)\times id\\
&= H(ev) \circ H(\Lambda f) \times H(id)\\
&= H(ev \circ \Lambda f \times id)\\
&= Hf,\\
\text{hence } j \circ H(\Lambda f) &= \Lambda(Hf).
\end{aligned}$$

Using these properties we will show that if $\Delta = x_1,\dots,x_n = \vec{x}$, $\vec{d} = d_1,\dots,d_n$ and $M \in \Lambda(|\mathfrak{M}|)$, then

$$H(\llbracket M\rrbracket_{\Delta})(*,\vec{d}) = \llbracket M^{\varphi}[\vec{x} := \underline{\vec{d}}]\rrbracket^{\mathfrak{N}}.$$

The proof is by induction on M (we skip the case $M \equiv x$):

$$\begin{aligned}
H(\llbracket \underline{a}\rrbracket_{\Delta})(*,\vec{d}) &= H(a \circ \Pi^{\Delta}_{\langle\,\rangle})(*,\vec{d}) \\
&= (Ha \circ \Pi^{\Delta}_{\langle\,\rangle})(*,\vec{d}) \\
&= Ha(*) \\
&= \varphi(a) \\
&= \llbracket \underline{a}^{\varphi}[\vec{x} := \underline{\vec{d}}]\rrbracket^{\mathfrak{N}}.
\end{aligned}$$

$$\begin{aligned}
H(\llbracket MN\rrbracket_{\Delta})(*,\vec{d}) &= H(ev \circ \langle F \circ \llbracket M\rrbracket_{\Delta},\llbracket N\rrbracket_{\Delta}\rangle)(*,\vec{d}) \\
&= (H(ev) \circ \langle HF \circ H\llbracket M\rrbracket_{\Delta},H\llbracket N\rrbracket_{\Delta}\rangle) (*,\vec{d}) \\
&= ev(\langle HF(H\llbracket M\rrbracket_{\Delta}(*,\vec{d})), H\llbracket N\rrbracket_{\Delta}(*,\vec{d})\rangle) \\
&= \llbracket M^{\varphi}[\vec{x} := \underline{\vec{d}}]\rrbracket^{\mathfrak{N}} \cdot \llbracket N^{\varphi}[\vec{x} := \underline{\vec{d}}]\rrbracket^{\mathfrak{N}} \\
&= \llbracket (MN)^{\varphi}[\vec{x} := \underline{\vec{d}}]\rrbracket^{\mathfrak{N}}.
\end{aligned}$$

Writing $(*,\vec{d}^{-})$ for $\Pi^{\Delta}_{\Delta\backslash x}(*,\vec{d})$ and $\vec{x}^{-}$ for $\Delta\backslash x$ we have

$$\begin{aligned}
H(\llbracket \lambda x.M\rrbracket_{\Delta})(*,\vec{d}) &= H(G \circ \Lambda(\llbracket M\rrbracket_{\Delta;x}) \circ \Pi^{\Delta}_{\Delta\backslash x})(*,\vec{d}) \\
&= HG \circ H\Lambda\llbracket M\rrbracket_{\Delta;x}(*,\vec{d}^{-}) \\
&= HG(\Lambda(H\llbracket M\rrbracket_{\Delta;x})(*,\vec{d}^{-})) \\
&= HG(e \mapsto H(\llbracket M\rrbracket_{\Delta;x})(*,\vec{d}^{-},e)) \\
&= HG(e \mapsto \llbracket M^{\varphi}[\vec{x}^{-} := \underline{\vec{d}}^{-}, x := \underline{e}]\rrbracket^{\mathfrak{N}}) \\
&= \llbracket \lambda x.M^{\varphi}[\vec{x}^{-} := \underline{\vec{d}}^{-}]\rrbracket^{\mathfrak{N}} \\
&= \llbracket (\lambda x.M)^{\varphi}[\vec{x} := \underline{\vec{d}}]\rrbracket^{\mathfrak{N}}.
\end{aligned}$$

Now we can finish our proof by noting that

$$\begin{aligned}
\varphi(\llbracket M\rrbracket^{\mathfrak{M}}) &= H(\llbracket M\rrbracket_{\langle\,\rangle})(*) \\
&= \llbracket M^{\varphi}\rrbracket^{\mathfrak{N}}.
\end{aligned}$$

□

Because of this theorem we will write $\mathfrak{M}(\mathcal{C})$ instead of $\mathfrak{M}(\mathcal{C},H)$.

2.5.7. Example. CPO is a strictly concrete category via the forgetful functor which assigns to every complete partial order its underlying set. In this category we may define several interesting categorical lambda models, such as

(1) $\mathbb{P}\omega = (P\omega, \cdot, \mathrm{fun}, \mathrm{graph})$, the graphmodel. See Scott [1976] and example 1.4.31.

(2) $\mathbb{D}_\infty = (D_\infty, \cdot, \varphi, \varphi^{-1})$. See Scott [1972] and example 1.5.6.

(3) $\mathbb{T}\omega = (T\omega, \cdot, \mathrm{fun}, \mathrm{graph})$. See Plotkin [1978] or Barendregt and Longo [1980].

□

CHAPTER 3

DERIVED LAMBDA ALGEBRAS AND THE CONSTRUCTION OF $\mathbb{D}_\infty$ INSIDE $\mathbb{P}\omega$

Using the theory of categorical interpretations as studied in chapter 2, we will show how to assign, under a certain basic assumption, to a retract $V \lhd U$ of a given model U an induced reflexive object structure such that it becomes a socalled *derived lambda algebra* of U. The theory of this derived model can be obtained by an associated translation of lambda terms.

An important example of this construction occurs when the considered retract $V \lhd U$ is the retract $U^U \lhd U$ of U onto its function space. In this case we talk about the canonical derived lambda algebra of U.

In the category of complete partial orders it is possible to take the limit of the iteration of this construction. When starting for instance with the lambda model $\mathbb{P}\omega$, we arrive in this way at an extensional derived model $\mathbb{P}_\infty \lhd \mathbb{P}\omega$, which is elementarily equivalent to the model $\mathbb{D}_\infty$. This technique of "constructing $\mathbb{D}_\infty$ inside $\mathbb{P}\omega$" was defined in Scott [1974], [1976].

3.1. *Derived lambda algebras in general.*

For the remainder of this section we fix a categorical lambda algebra $\mathcal{C} = (C,U,F,G)$.

3.1.1. <u>Definition</u>. Let $(g,f)\colon A \lhd B$ be a retract in C.
Define $\bar{g}\colon A^A \to B^B$ and $\bar{f}\colon B^B \to A^A$ by

$$\bar{g} = \Lambda(g \circ ev \circ id \times f) \quad\text{and}\quad \bar{f} = \Lambda(f \circ ev \circ id \times g).$$

□

Note that $\bar{f}$, resp. $\bar{g}$, depend on the pair (g,f) and not only on f, resp. g. A more precise, but also more cumbersome, notation for $\bar{g}$ would be g^f.

Similarly $\bar{f}$ would become f^{g}.

3.1.2. Lemma. Let $(g,f)\colon A \lhd B$ be a retract in $\mathcal{C}$.
Then so is $(\bar{g},\bar{f})\colon A^{A} \lhd B^{B}$.

Proof: Calculate

$$\begin{aligned}
\bar{f} \circ \bar{g} &= \Lambda(f \circ ev \circ id \times g) \circ \bar{g} \\
&= \Lambda(f \circ ev \circ id \times g \circ \bar{g} \times id) \\
&= \Lambda(f \circ ev \circ \bar{g} \times id \circ id \times g) \\
&= \Lambda(f \circ g \circ ev \circ id \times f \circ id \times g) \\
&= \Lambda(ev) \\
&= id_{A^{A}}.
\end{aligned}$$

□

3.1.3. Remark. The assignment $(g,f) \mapsto (\bar{g},\bar{f})$ is functorial in the following sense.

Let $Ret(\mathcal{C})$ be the category with

objects: retracts $(g,f)\colon A \lhd B$,

morphisms (from $(g,f)\colon A \lhd B$ to $(g',f')\colon A' \lhd B'$):

pairs of retracts $(p,p')\colon A \lhd A'$

and $(q,q')\colon B \lhd B'$

such that

$$\begin{tikzcd}
A \arrow[r, shift left, "g"] \arrow[d, shift right, "p"'] & B \arrow[l, shift left, "f"] \arrow[d, shift right, "q"'] \\
A' \arrow[r, shift left, "g'"] \arrow[u, shift right, "p'"'] & B' \arrow[l, shift left, "f'"] \arrow[u, shift right, "q'"']
\end{tikzcd}$$

$$q \circ g = g' \circ p \quad \text{and} \quad f \circ q' = p' \circ f'.$$

Then $H(g,f) = (\bar{g},\bar{f})$ (for objects and both components of morphisms) defines an endofunctor $Ret(\mathcal{C}) \to Ret(\mathcal{C})$.

□

For the remainder of this section, fix a retract $(r,s)\colon V \lhd U$. We will try to induce a reflexive structure on the object V as follows.

3.1.4. Definition. Define $F': V \to V^V$ and $G': V^V \to V$ by

$F' = \bar{s} \circ F \circ r$ and $G' = s \circ G \circ \bar{r}$

$$\begin{tikzcd}
U^U \arrow[r, shift left, "G"] \arrow[d, shift right, "\bar{s}"'] & U \arrow[l, shift left, "F"] \arrow[d, shift right, "s"'] \\
V^V \arrow[r, shift left, "G'"] \arrow[u, shift right, "\bar{r}"'] & V \arrow[l, shift left, "F'"] \arrow[u, shift right, "r"'] .
\end{tikzcd}$$

□

We are interested in the case that this definition turns V into a reflexive object.

3.1.5. Definition.

(i) The following statement is called the *basic assumption*:

(BA) $(G',F'): V^V \lhd V$.

(ii) If (BA) is satisfied, then $(G',F'): V^V \lhd V$ induces a lambda algebra, the socalled *derived lambda algebra* of the lambda algebra $(G,F): U^U \lhd U$ (with respect to C and $(r,s): V \lhd U$).

□

There is an important case in which (BA) holds.

3.1.6. Example. If we take $V = U^U$, $r = G$, $s = F$ then $F' = \bar{F} \circ F \circ G = \bar{F}$ and $G' = F \circ G \circ \bar{G} = \bar{G}$. Hence by lemma 3.1.2. (BA) is satisfied.

The lambda algebra induced in this way is called the *canonical derived lambda algebra.*

□

The aim of the remainder of this section is to give a necessary and sufficient condition for (BA) to hold and to establish a relationship (via translation) between the original model and the derived model (with respect to $(r,s): V \lhd U$).

We remind the reader of the useful retract $(\alpha,\beta)\colon \mathcal{C}(U,U) \lhd \mathcal{C}(T,U)$, defined in 2.4.1.

3.1.7. <u>Definition</u>.

(i) $\delta = r \circ s \in \mathcal{C}(U,U)$.

(ii) $d = \alpha(\delta) \in \mathcal{C}(T,U)$.

□

To distinguish between the original model and the derived model we let $\mathfrak{M} = \mathfrak{M}(\mathcal{C},U,F,G)$ be the original lambda algebra and (if (BA) is satisfied) we let $\mathfrak{N} = \mathfrak{M}(\mathcal{C},V,F',G')$ be the derived lambda algebra. From now on we will use P,Q,R,... as metavariables for lambda terms.

3.1.8. <u>Lemma</u>. Let $a,b \in |\mathfrak{M}|$, $P \in \Lambda(\mathfrak{M})$, $|\Delta| \supseteq FV(P)$.

(i) $[\![\underline{a}P]\!]^{\mathfrak{M}}_{\Delta} = \beta(a) \circ [\![P]\!]^{\mathfrak{M}}_{\Delta}$.

In particular

$$[\![\underline{d}P]\!]^{\mathfrak{M}}_{\Delta} = \delta \circ [\![P]\!]^{\mathfrak{M}}_{\Delta}.$$

(ii) $a \bullet b = \beta(a) \circ b$.

(iii) $a \circ b = \alpha(\beta(a) \circ \beta(b))$.

Proof:

(i) $[\![\underline{a}P]\!]^{\mathfrak{M}}_{\Delta} = [\![\underline{a}x]\!]_{x} \circ [[\![P]\!]^{\mathfrak{M}}_{\Delta}]$, by lemma 2.1.5.,

$= [\![\underline{a}x]\!]_{x} \circ [id] \circ [\![P]\!]^{\mathfrak{M}}_{\Delta}$

$= \beta(a) \circ [\![P]\!]^{\mathfrak{M}}_{\Delta}$, by definition 2.4.1.

Furthermore $\beta(d) = \beta(\alpha(\delta)) = \delta$.

(ii) $a \bullet b = [\![\underline{a}\underline{b}]\!]^{\mathfrak{M}}_{\langle\rangle}$

$= \beta(a) \circ [\![\underline{b}]\!]^{\mathfrak{M}}_{\langle\rangle}$, by (i),

$= \beta(a) \circ b$.

(iii) $a \circ b = [\![\lambda x.\ \underline{a}(\underline{b}x)]\!]^{\mathfrak{M}}_{\langle\rangle}$

$= G \circ \Lambda([\![\underline{a}(\underline{b}x)]\!]^{\mathfrak{M}}_{x})$

$= G \circ \Lambda(\beta(a) \circ \beta(b) \circ [\![x]\!]^{\mathfrak{M}}_{x})$, by (i),

$= \alpha(\beta(a) \circ \beta(b))$, by definition 2.4.1.

□

We now have gathered enough material to characterize the basic assumption, completely in terms of $\mathfrak{M}$.

3.1.9. <u>Theorem</u>. (BA) $\Longleftrightarrow$ $\mathfrak{M} \models d^{d} \circ d \circ d^{d} = d^{d}$.

Proof: Let us introduce the following abbreviations:

$$H = F \circ \delta \circ G\colon U^{U} \to U^{U},$$

$$D = \delta \circ \mathrm{ev} \circ \mathrm{id} \times \delta\colon U^{U} \times U \to U,$$

$$R = [\![d^{d}xy]\!]_{x,y} \quad , \quad L = [\![d^{d}(d(d^{d}x))y]\!]_{x,y}.$$

Note that the right-hand-side of the equation $d^{d} \circ d \circ d^{d} = d^{d}$ can be written as $G \circ \Lambda(G \circ \Lambda(R))$ and similarly the left-hand-side as $G \circ \Lambda(G \circ \Lambda(L))$.

In order to compute the values of R and L, we have to calculate successively:

$$[\![dy]\!]_{x,y} = \delta \circ q \quad \text{, by lemma 3.1.8.(i).}$$

$$\begin{aligned} [\![x(dy)]\!]_{x,y} &= \mathrm{ev} \circ \langle F \circ q \circ p,\ \delta \circ q \rangle \\ &= \mathrm{ev} \circ ((F \circ q) \times \delta). \end{aligned}$$

$$[\![d(x(dy))]\!]_{x,y} = \delta \circ \mathrm{ev} \circ ((F \circ q) \times \delta),$$

hence $\qquad R = D \circ ((F \circ q) \times \mathrm{id}) \qquad (1).$

And also
$$\begin{aligned} [\![\lambda y.\ d(x(dy))]\!]_{x} &= G \circ \Lambda(R). \\ [\![d(d^{d}x)]\!]_{x,y} &= \delta \circ G \circ \Lambda(R) \circ p. \\ [\![d(d(d^{d}x)(dy))]\!]_{x,y} &= \delta \circ \mathrm{ev} \circ \langle F \circ \delta \circ G \circ \Lambda(R) \circ p,\ \delta \circ q \rangle \\ &= \delta \circ \mathrm{ev} \circ \mathrm{id} \times \delta \circ ((H \circ \Lambda(R)) \times \mathrm{id}), \end{aligned}$$

so $\qquad L = D \circ ((H \circ \Lambda(R)) \times \mathrm{id}) \qquad (2).$

Furthermore we will need
$$\begin{aligned} \bar{r} \circ \bar{s} &= \Lambda(r \circ \mathrm{ev} \circ \mathrm{id} \times s \circ \bar{s} \times \mathrm{id}) \\ &= \Lambda(r \circ \mathrm{ev} \circ \bar{s} \times \mathrm{id} \circ \mathrm{id} \times s) \\ &= \Lambda(r \circ s \circ \mathrm{ev} \circ \mathrm{id} \times r \circ \mathrm{id} \times s) \end{aligned}$$

$$= \Lambda(\delta \circ \mathrm{ev} \circ \mathrm{id} \times \delta),$$

so $$\bar{r} \circ \bar{s} = \Lambda(D) \qquad (3).$$

Now everything can be put together to give

$$\begin{aligned}
(\mathrm{BA}) \;&\Longleftrightarrow\; F' \circ G' = \mathrm{id} \\
&\Longleftrightarrow\; \bar{s} \circ F \circ r \circ s \circ G \circ \bar{r} = \mathrm{id} \\
&\Longleftrightarrow\; \bar{s} \circ H \circ \bar{r} = \mathrm{id} \\
&\Longleftrightarrow\; \Lambda(s \circ \mathrm{ev} \circ (\mathrm{id} \times r) \circ ((H \circ \bar{r}) \times \mathrm{id})) = \mathrm{id} \\
&\Longleftrightarrow\; s \circ \mathrm{ev} \circ ((H \circ \bar{r}) \times r) = \mathrm{ev} \\
&\Longleftrightarrow\; r \circ s \circ \mathrm{ev} \circ ((H \circ \bar{r}) \times r) \circ (\bar{s} \times s) = r \circ \mathrm{ev} \circ \bar{s} \times s \\
&\Longleftrightarrow\; \delta \circ \mathrm{ev} \circ ((H \circ \bar{r} \circ \bar{s}) \times \delta) = r \circ s \circ \mathrm{ev} \circ (\mathrm{id} \times r) \circ (\mathrm{id} \times s) \\
&\overset{(3)}{\Longleftrightarrow}\; D \circ ((H \circ \Lambda(D)) \times \mathrm{id}) = D \\
&\overset{(*)}{\Longleftrightarrow}\; D \circ ((H \circ \Lambda(D) \circ F \circ q) \times \mathrm{id}) = D \circ ((F \circ q) \times \mathrm{id}) \\
&\overset{(1)}{\Longleftrightarrow}\; D \circ ((H \circ \Lambda(R)) \times \mathrm{id}) = R \\
&\overset{(2)}{\Longleftrightarrow}\; L = R \\
&\Longleftrightarrow\; \mathfrak{M} \models d^d \circ d \circ d^d = d^d.
\end{aligned}$$

The equivalence labeled (*) holds, because $(F \circ q) \times \mathrm{id}$ is (split) epi with right inverse $\langle !, G\rangle \times \mathrm{id}$.

□

From now on, if we talk about the lambda algebra $\mathfrak{N}$, it is tacitly understood that the basic assumption holds.

The relationship between the interpretation in the derived structure $\mathfrak{N}$ and the original lambda algebra $\mathfrak{M}$ can be given by the following translation.

3.1.10. Definition. A translation $P \mapsto P^d \colon \Lambda(\mathfrak{N}) \to \Lambda(\mathfrak{M})$ is inductively defined by

$$\begin{aligned}
x^d &= \underline{d}x \\
\underline{c}^d &= \underline{r \circ c} \quad , \text{ for } c \in |\mathfrak{N}| = C(T,V), \\
(PQ)^d &= \underline{d}(P^d Q^d) \\
(\lambda x.P)^d &= \underline{d}(\lambda x.P^d).
\end{aligned}$$

□

An immediate consequence of this definition is the fact that $FV(P^d) = FV(P)$ for all $P \in \Lambda(\mathfrak{N})$.

Recall the definition of $(A^n)_{n\in\omega}$ in 2.1.1.(ii). Now, if $f\colon A \to B$, we naturally define by induction on $n \in \omega$ mappings $f^n\colon A^n \to B^n$ as:

$$f^0 = id_T \quad , \quad f^{n+1} = f^n \times f .$$

3.1.11. <u>Proposition</u>. Let $\Delta = x_1,\ldots,x_n \supseteq FV(P)$. Then

$$[\![P^d]\!]^{\mathfrak{M}}_{\Delta} = r \circ [\![P]\!]^{\mathfrak{N}}_{\Delta} \circ s^n .$$

Proof: Using lemma 3.1.8.(i) several times, the proof proceeds by induction on the structure of P.

$$\begin{aligned}
[\![x^d]\!]^{\mathfrak{M}}_{\Delta} &= [\![\underline{d}x]\!]^{\mathfrak{M}}_{\Delta} \\
&= \delta \circ [\![x]\!]^{\mathfrak{M}}_{\Delta} \\
&= r \circ s \circ (\pi^{\Delta}_{x})^{\mathfrak{M}} \\
&= r \circ (\pi^{\Delta}_{x})^{\mathfrak{N}} \circ s^n \\
&= r \circ [\![x]\!]^{\mathfrak{N}}_{\Delta} \circ s^n \quad ,
\end{aligned}$$

$$\begin{aligned}
[\![\underline{c}^d]\!]^{\mathfrak{M}}_{\Delta} &= [\![\underline{r \circ c}]\!]^{\mathfrak{M}}_{\Delta} \\
&= r \circ c \circ !_{U^n} \\
&= r \circ c \circ !_{V^n} \circ s^n \\
&= r \circ [\![\underline{c}]\!]^{\mathfrak{N}}_{\Delta} \circ s^n \quad ,
\end{aligned}$$

$$\begin{aligned}
[\![(PQ)^d]\!]^{\mathfrak{M}}_{\Delta} &= [\![\underline{d}(P^d Q^d)]\!]^{\mathfrak{M}}_{\Delta} \\
&= \delta \circ ev_{UU} \circ \langle F \circ [\![P^d]\!]^{\mathfrak{M}}_{\Delta}, [\![Q^d]\!]^{\mathfrak{M}}_{\Delta} \rangle \\
&= r \circ s \circ ev_{UU} \circ \langle F \circ r \circ [\![P]\!]^{\mathfrak{N}}_{\Delta}, r \circ [\![Q]\!]^{\mathfrak{N}}_{\Delta} \rangle \circ s^n \text{ , by IH,} \\
&= r \circ ev_{VV} \circ \bar{s} \times id \circ \langle F \circ r \circ [\![P]\!]^{\mathfrak{N}}_{\Delta}, [\![Q]\!]^{\mathfrak{N}}_{\Delta} \rangle \circ s^n \\
&= r \circ ev_{VV} \circ \langle F' \circ [\![P]\!]^{\mathfrak{N}}_{\Delta}, [\![Q]\!]^{\mathfrak{N}}_{\Delta} \rangle \circ s^n \\
&= r \circ [\![PQ]\!]^{\mathfrak{N}}_{\Delta} \circ s^n .
\end{aligned}$$

For the last step in this induction we introduce the following abbreviations:

$$\Pi = (\Pi^{\Delta}_{\Delta\backslash x})^{\mathfrak{M}}\ ,\ \Pi' = (\Pi^{\Delta}_{\Delta\backslash x})^{\mathfrak{N}}\ ;\ k = \#(\Delta\backslash x).$$

Then

$$\begin{aligned}
\llbracket (\lambda x.P)^{d} \rrbracket^{\mathfrak{M}}_{\Delta} &= \llbracket \underline{d}(\lambda x.P^{d}) \rrbracket^{\mathfrak{M}}_{\Delta} \\
&= \delta \circ G \circ \Lambda(\llbracket P^{d} \rrbracket^{\mathfrak{M}}_{\Delta;x}) \circ \Pi \\
&= \delta \circ G \circ \Lambda(r \circ \llbracket P \rrbracket^{\mathfrak{N}}_{\Delta;x} \circ (s^{k} \times s)) \circ \Pi\ ,\ \text{by IH}, \\
&= \delta \circ G \circ \Lambda(r \circ \llbracket P \rrbracket^{\mathfrak{N}}_{\Delta;x} \circ id \times s) \circ s^{k} \circ \Pi \\
&= \delta \circ G \circ \Lambda(r \circ ev_{VV} \circ \Lambda(\llbracket P \rrbracket^{\mathfrak{N}}_{\Delta;x}) \times s) \circ \Pi' \circ s^{n} \\
&= r \circ s \circ G \circ \bar{r} \circ \Lambda(\llbracket P \rrbracket^{\mathfrak{N}}_{\Delta;x}) \circ \Pi' \circ s^{n} \\
&= r \circ G' \circ \Lambda(\llbracket P \rrbracket^{\mathfrak{N}}_{\Delta;x}) \circ \Pi' \circ s^{n} \\
&= r \circ \llbracket \lambda x.P \rrbracket^{\mathfrak{N}}_{\Delta} \circ s^{n}\ .
\end{aligned}$$

□

3.1.12. <u>Theorem</u>. Let $P,Q \in \Lambda(\mathfrak{N})$, $\Delta = x_1,\dots,x_n \supseteq FV(PQ)$.

(i) $\llbracket P \rrbracket^{\mathfrak{N}}_{\Delta} = s \circ \llbracket P^{d} \rrbracket^{\mathfrak{M}}_{\Delta} \circ r^{n}$.

(ii) $\mathfrak{N} \models P = Q \iff \mathfrak{M} \models P^{d} = Q^{d}$, for $P,Q \in \Lambda^{0}(\mathfrak{N})$.

Proof: Immediate by proposition 3.1.11.

□

The construction of the derived model $\mathfrak{N}$, as we introduced it, depends on the representation of the original model $\mathfrak{M}$ as a categorical lambda algebra (C,U,F,G) and on the retract $(r,s)\colon V \lhd U$ in C. But in fact, as we will show now, the derived model $\mathfrak{N}$ depends only on the structure of $\mathfrak{M}$ as a lambda algebra and on the element $d \in |\mathfrak{M}|$, representing the retract $(r,s)\colon V \lhd U$.

3.1.13. <u>Lemma</u>. There exists a retract $(\rho,\sigma)\colon |\mathfrak{N}| \lhd |\mathfrak{M}|$ defined by

$$\rho(y) = r \circ y,$$
$$\sigma(x) = s \circ x.$$

Proof: $\sigma(\rho(y)) = s \circ r \circ y = y$.

□

Now it is easy to transport the lambda algebra structure from $|\mathfrak{N}|$ to $\rho(|\mathfrak{N}|)$ via the injection ρ.

3.1.14. <u>Proposition</u>.

(i) $\rho(|\mathfrak{N}|) = \mathrm{Fix}(d) = \{x \in |\mathfrak{M}| \mid d \cdot x = x\}$.

(ii) For all $a,b \in |\mathfrak{N}|$ we have

$$\rho(a \cdot_{\mathfrak{N}} b) = d(\rho(a)\rho(b)).$$

(iii) For all $P \in \Lambda^0(\mathfrak{N})$ we have

$$\rho([\![P]\!]^{\mathfrak{N}}) = [\![P^d]\!]^{\mathfrak{M}}.$$

Proof:

(i) $$\begin{aligned} x \in \rho(|\mathfrak{N}|) &\Longleftrightarrow \exists y \in |\mathfrak{N}| \ (x = r \circ y) \\ &\Longleftrightarrow \exists y \in |\mathfrak{N}| \ (x = r \circ y \wedge y = s \circ x) \\ &\Longleftrightarrow x = r \circ s \circ x = \delta \circ x \\ &\Longleftrightarrow x = dx. \end{aligned}$$

(ii) $$\begin{aligned} \rho(a \cdot_{\mathfrak{N}} b) &= r \circ [\![\underline{a}\underline{b}]\!]^{\mathfrak{N}} \\ &= [\![(\underline{a}\underline{b})^d]\!]^{\mathfrak{M}} \quad \text{, by proposition 3.1.11.,} \\ &= d(\rho(a)\rho(b)). \end{aligned}$$

(iii) follows immediately from proposition 3.1.11.

□

Now we will state in a definition the properties of $d \in |\mathfrak{M}|$ needed to produce a derived model in general, not depending on any representation of $\mathfrak{M}$ as a categorical lambda algebra C.

3.1.15. <u>Definition</u>. Let $\mathfrak{M}$ be a lambda algebra. An element $d \in |\mathfrak{M}|$ is called *derivable* if

$$d \circ d = d \quad \text{and} \quad d^d \circ d \circ d^d = d^d.$$

□

The fact that these two properties completely determine the retracts of the reflexive object in a categorical lambda algebra, that give rise to a new reflexive object, can be stated as follows.

3.1.16. Theorem. Let $\mathfrak{M}$ be a lambda algebra.

(i) A derived model $\mathfrak{N}$ of $\mathfrak{M}$, represented as a categorical lambda algebra $\mathcal{C}$, is completely determined (up to isomorphism) by the element $d \in |\mathfrak{M}|$ representing the retract $(r,s)\colon V \lhd U$. This element d is derivable.

(ii) Every derivable $d \in |\mathfrak{M}|$ corresponds to some derived model $\mathfrak{N}$. To be more specific: $\mathfrak{N}$ is the derived model induced by the retract $(d,d)\colon d \lhd I$ in $\mathcal{K}(\mathfrak{M})$.

Proof:

(i) The first statement clearly follows from proposition 3.1.14. The fact that d is derivable follows from theorem 3.1.9. (this shows $d^d \circ d \circ d^d = d^d$) and lemma 3.1.8.(iii):

$$
\begin{aligned}
d \circ d &= \alpha(\beta(d) \circ \beta(d)) \\
&= \alpha(\delta \circ \delta) \\
&= \alpha(\delta) \qquad \text{, since } \delta \text{ is idempotent,} \\
&= d.
\end{aligned}
$$

(ii) Let $d \in |\mathfrak{M}|$ be derivable. Since $d \circ d = d$ we know that d is an object in $\mathcal{K}(\mathfrak{M})$. Therefore we can draw the following diagram

$$
\begin{tikzcd}
U^U = 1 \arrow[r, shift left, "G = 1"] & I = U \arrow[l, shift left, "F = 1"] \arrow[d, shift right, "s = d"'] \\
 & d = V \arrow[u, shift right, "d = r"']
\end{tikzcd}
$$

.

Then $\delta = r \circ s = d \circ d = d$ and

$$
\begin{aligned}
\alpha(\delta) &= G \circ \Lambda(\delta \circ q_{TU}) \\
&= 1 \circ \Lambda(d \circ q) \\
&= \lambda xy.\ d(q[x,y]) \\
&= \lambda xy.\ dy \\
&= kd.
\end{aligned}
$$

Therefore $\alpha(\delta) = kd \in K(\mathfrak{M})(T,I)$ corresponds exactly to $d \in |\mathfrak{M}|$ under the isomorphism $\mathfrak{M}(K(\mathfrak{M})) \cong \mathfrak{M}$ of theorem 2.4.10.

Now since $d^d \circ d \circ d^d = d^d$ in $\mathfrak{M}$, it follows by theorem 3.1.9. that the basic assumption is satisfied and hence that the above diagram defines a derived lambda algebra.

□

In many cases it is easier to consider $\rho(\mathfrak{N})$ as the derived lambda algebra instead of $\mathfrak{N}$. Since $\mathfrak{N}$ and $\rho(\mathfrak{N})$ are isomorphic we will not formally distinguish between them. In this way the construction is quite simple, as we will show now.

3.1.17. Corollary. Let $\mathfrak{M}$ be a lambda algebra and $d \in |\mathfrak{M}|$ a derivable element. The derived lambda algebra induced by d, notation $\mathfrak{M}(d) = (|\mathfrak{M}(d)|, \cdot_d, [\![\cdot]\!]^d)$, can be described as follows:

(a) $|\mathfrak{M}(d)| = \mathrm{Fix}(d)$;

(b) $a \cdot_d b = d(ab)$ for all $a,b \in \mathrm{Fix}(d)$;

(c) For all $P \in \Lambda^0(\mathfrak{M}(d))$ we have $[\![P]\!]^d = [\![P^d]\!]$, where the translation $P \mapsto P^d : \Lambda(\mathfrak{M}(d)) \to \Lambda(\mathfrak{M})$ is defined by

$$\begin{aligned} x^d &= \underline{d}x \\ \underline{c}^d &= \underline{c} \\ (PQ)^d &= \underline{d}(P^dQ^d) \\ (\lambda x.P)^d &= \underline{d}(\lambda x.P^d). \end{aligned}$$

□

For future use in section 3 we now show how to construct new derivable elements from old ones.

3.1.18. Lemma. Let $\mathfrak{M}$ be a lambda algebra.
If $d \in |\mathfrak{M}|$ is derivable then so are d^d and $d \circ d^d \circ d$.

Proof: Let d be derivable. Then we have the following diagram in $K(\mathfrak{M})$:

$$\begin{tikzcd}
1 \arrow[r, shift left, "1"] \arrow[d, shift right, "d^d"'] & I \arrow[l, shift left, "1"] \arrow[d, shift right, "d"'] \\
d^d \arrow[u, shift right, "d^d"'] \arrow[r, shift left, "d \circ d^d"] & d \arrow[u, shift right, "d"'] \arrow[l, shift left, "d^d \circ d"] \arrow[d, shift right, "d^d \circ d"'] \\
 & d^d \arrow[u, shift right, "d \circ d^d"']
\end{tikzcd}$$

We note here that $d^d \in |K(\mathfrak{M})|$ and hence $d^d \circ d^d = d^d$.

By taking the canonical derived model of the first derived model we get by composition (since the operation $(g,f) \mapsto (\bar{g},\bar{f})$ is functorial by remark 3.1.3.) a derived model of the form

$$\begin{tikzcd}
1 \arrow[r, shift left, "1"] & I \arrow[l, shift left, "1"] \arrow[d, shift right, "d^d \circ d"'] \\
 & d^d \arrow[u, shift right, "d \circ d^d"']
\end{tikzcd}$$

The derivable element corresponding to this derived model is $d \circ d^d \circ d^d \circ d = d \circ d^d \circ d$.

To show that d^d is derivable there seems to be no other way than by direct calculation:

We already saw that $d^d \circ d^d = d^d$.

In general
$$\begin{aligned} a_1^{b_1} \circ a_2^{b_2} &= \lambda x . a_1^{b_1}(a_2^{b_2} x) \\ &= \lambda x . a_1 \circ (a_2 \circ x \circ b_2) \circ b_1 \\ &= \lambda x . (a_1 \circ a_2) \circ x \circ (b_2 \circ b_1) \\ &= (a_1 \circ a_2)^{b_2 \circ b_1} . \end{aligned}$$

$$\text{Therefore } (d^d)^{d^d} \circ d^d \circ (d^d)^{d^d} = (d^d \circ d \circ d^d)^{d^d \circ d \circ d^d}$$
$$= (d^d)^{d^d}.$$

Hence d^d is indeed derivable.

□

3.1.19. Examples.

(i) For every $n \in \omega$ we have the derivable element

$$1_n = \lambda xy_1 \ldots y_n . xy_1 \ldots y_n .$$

(ii) As an example of a derivable element d that is not an η-expansion of I we may take

$$d = \lambda xz.z(\lambda y.x(\lambda w.wy)) ,$$

which can be written as

$$d = \lambda x.\langle \lambda y.x\langle y\rangle\rangle$$

if we put $\langle P\rangle = \lambda z.zP$ for a fresh variable z.

(iii) For any derivable d we can construct more derivable elements by iteration, using lemma 3.1.18.

□

The last part of this section describes what kind of extensionality properties a derived model has.

3.1.20. Proposition. Let $\mathfrak{M}$ be a lambda algebra and $d \in |\mathfrak{M}|$ a derivable element. Then

(a) $\mathfrak{M}(d) \models 1 = I \iff \mathfrak{M} \models d \circ d^d \circ d = d$,

(b) $\mathfrak{M}(d)$ is weakly extensional $\iff$

$$\forall ab \in |\mathfrak{M}| \; (\forall x \in |\mathfrak{M}| (ax = bx) \rightarrow d^d a = d^d b).$$

Proof: We consider the following diagram in $K(\mathfrak{M})$

$$\begin{tikzcd}
1 \arrow[r, shift left, "1"] \arrow[d, shift right, "d^d"'] & I \arrow[l, shift left, "1"] \arrow[d, shift right, "d"'] \\
d^d \arrow[u, shift right, "d^d"'] \arrow[r, shift left, "d \circ d^d"] & d \arrow[l, shift left, "d^d \circ d"] \arrow[u, shift right, "d"']
\end{tikzcd} .$$

(a) $\mathfrak{M}(d) \models 1 = I \iff (d \circ d^d) \circ (d^d \circ d) = d$, by proposition 2.1.13.(ii),

$\iff d \circ d^d \circ d = d.$

(b) $\mathfrak{M}(d)$ is weakly extensional

$\iff K(\mathfrak{M})$ has enough points at d, by proposition 2.1.13.(i),

$\iff \forall f,g\colon d \to d(\forall x\colon T \to d(f \circ x = g \circ x) \to f = g)$

$\iff \forall ab \in |\mathfrak{M}|\ (\forall x \in |\mathfrak{M}|(d \circ a \circ d \circ K(dx) = d \circ b \circ d \circ K(dx)) \to$
$d \circ a \circ d = d \circ b \circ d)$

$\iff \forall ab \in |\mathfrak{M}|\ (\forall x \in |\mathfrak{M}|(K(d^d ax) = K(d^d bx)) \to d^d a = d^d b)$

$\iff \forall ab \in |\mathfrak{M}|\ (\forall x \in |\mathfrak{M}|(ax = bx) \to d^d a = d^d b).$

□

3.1.21. Corollary. Let $\mathfrak{M}$ be a lambda algebra and $d \in |\mathfrak{M}|$ a derivable element.

(i) If $d^d = d$ then $\mathfrak{M}(d) \models 1 = I$.

(ii) If $\mathfrak{M}$ is a lambda model then so is $\mathfrak{M}(d)$.

Proof:

(i) By proposition 3.1.20.(a).

(ii) Suppose $a,b \in |\mathfrak{M}|$ and $\forall x \in |\mathfrak{M}|(a.x = b.x)$. Since $\mathfrak{M}$ is weakly extensional we then have $1a = 1b$. Now note that $d^d \circ 1 = d^d \circ I^I = (d \circ I)^{I \circ d} = d^d$. Hence $d^d a = d^d(1a) = d^d(1b) = d^d b$. Applying proposition 3.1.20.(b) we conclude that $\mathfrak{M}(d)$ is weakly extensional.

□

Since the property $d^d = d$ will play an important role in section 3.4. we will supply a name for it.

3.1.22. Definition. Let $\mathfrak{M}$ be a lambda algebra. An element $d \in |\mathfrak{M}|$ is called *strongly extensional* if $d = d \circ d = d^d$.

□

Clearly any strongly extensional element is derivable. Models derived from strongly extensional elements have the nice property, that the translation of definition 3.1.10. commutes with application and abstraction.

3.1.23. Lemma. Let $\mathfrak{M}$ be a lambda algebra and let $d \in |\mathfrak{M}|$ be strongly extensional.

(i) $|\mathfrak{M}(d)|$ is closed under application (in $\mathfrak{M}$).
Hence $\mathfrak{M} \models (PQ)^d = P^dQ^d$, for all $P,Q \in \Lambda^0(\mathfrak{M}(d))$.

(ii) $\mathfrak{M} \models (\lambda x.P)^d = \lambda x.P^d$, for all $P \in \Lambda^0(\mathfrak{M}(d),x)$.

Proof:

(i) Suppose $a,b \in |\mathfrak{M}(d)| = \mathrm{Fix}(d)$. Then $da = a$, $db = b$.

$$\begin{aligned} \text{Hence}\quad ab &= dab \\ &= d^d ab \quad \text{, since } d \text{ is strongly extensional,} \\ &= d(a(db)) \\ &= d(ab) \in |\mathfrak{M}(d)|. \end{aligned}$$

The second statement follows, because $P^d,Q^d \in |\mathfrak{M}(d)|$ and so $(PQ)^d = d(P^dQ^d) = P^dQ^d$ in $\mathfrak{M}$.

(ii) Calculate in $\mathfrak{M}$:
It is easy to show by induction on P that

$$\lambda x.dP^d[x := dx] = \lambda x.P^d.$$

$$\begin{aligned} \text{Then } (\lambda x.P)^d &= d(\lambda x.P^d) \\ &= d^d(\lambda x.P^d) \\ &= \lambda x.d((\lambda x.P^d)(dx)) \\ &= \lambda x.dP^d[x := dx] \\ &= \lambda x.P^d. \end{aligned}$$

□

3.1.24. Corollary. Let $\mathfrak{M}$ be a lambda algebra and let $d \in |\mathfrak{M}|$ be strongly extensional. Then $[\![P]\!]^d = [\![P \restriction d]\!]$, where $P \restriction d$ equals P with all variables x replaced by dx.

Proof: By lemma 3.1.23.

□

3.2. *Models inside the Karoubi-envelope of a lambda algebra.*

As shown in section 3.1., derived models in a general category C can always be represented as derived models in the category $K(\mathfrak{M})$, where $\mathfrak{M}$ is the original model.

It is therefore appropriate to look at (arbitrary) models defined inside $K(\mathfrak{M})$.

3.2.1. Convention. For the remainder of this section, let $\mathfrak{M}$ be a lambda algebra, $K(\mathfrak{M})$ its Karoubi-envelope and

$$\begin{tikzcd} b^b \arrow[r, shift left, "g"] & b \arrow[l, shift left, "f"] \end{tikzcd}$$

a retract in $K(\mathfrak{M})$, defining a lambda algebra $\mathfrak{N}$.

□

Note that $\mathfrak{N}$ doesn't need to be a derived model of $\mathfrak{M}$. Without further qualifications, notations like $\cdot, \circ$ etc. refer to application, composition etc. in $\mathfrak{M}$, not in $\mathfrak{N}$.

Now we will express the interpretation in $\mathfrak{N}$ in terms of $\mathfrak{M}$.

3.2.2. Definition. A translation $' : \Lambda(\mathfrak{N}) \to \Lambda(\mathfrak{M})$ is defined by induction on $P \in \Lambda(\mathfrak{N})$,

$$\begin{aligned} x' &= \underline{b}x \\ \underline{c}' &= \underline{cI} \quad , \text{ if } c \in |\mathfrak{N}| = K(T,b), \\ (PQ)' &= \underline{f}P'Q' \\ (\lambda x.P)' &= \underline{g}(\lambda x.P'). \end{aligned}$$

□

3.2.3. Theorem.

(i) $|\mathfrak{N}| = K(T,b) = \{Kc \mid bc = c\}$.

(ii) For any $P \in \Lambda(\mathfrak{N})$ such that $|\Delta| \supseteq FV(P)$, $\Delta = x_1,\dots,x_n$ we have

$$[\![P]\!]^{\mathfrak{N}}_{\Delta} = \lambda z.P'[x_n := z(KI),\dots,x_1 := zK^{\sim(n-1)}(KI)].$$

(iii) If $bc_1 = c_1$, $bc_2 = c_2$ then $(Kc_1) \cdot_{\mathfrak{N}} (Kc_2) = K(fc_1c_2)$.

Proof:

(i) $$\begin{aligned} K(T,b) &= \{m \mid b \circ m \circ KI = m\} \\ &= \{m \mid \lambda x.b(mI) = m\} \\ &= \{Kc \mid K(b(KcI)) = Kc\} \\ &= \{Kc \mid bc = c\}. \end{aligned}$$

(ii) As a basis step for a proof by induction on P it will be shown that

$$[\![x_{n-i}]\!]^{\mathfrak{N}}_{\Delta} = \lambda x.b(xK^{\sim i}(KI)), \quad 0 \leqslant i < n.$$

For $[\![x_{n-i}]\!]^{\mathfrak{N}}_{\Delta} = \pi^{\Delta}_{x_{n-i}}$

$$\overset{(*)}{=} b \circ q \circ \overbrace{p \circ \ldots \circ p}^{i \text{ times}}$$

$$= \lambda x.b(q(p^{i}x))$$

$$= \lambda x.b(q(xK^{\sim i}))$$

$$= \lambda x.b(xK^{\sim i}(KI)) .$$

(*): We used the fact that in general

$$p_{a_1,a_2} \circ p_{a_1 \times a_2,a_3} = p_{a_1,a_2} \circ (a_1 \times a_2) \circ p$$

$$= p_{a_1,a_2} \circ p, \text{ since } \mathrm{dom}(p_{a_1,a_2}) = a_1 \times a_2,$$

and similarly $q_{a_1,a_2} \circ p_{a_1 \times a_2,a_3} = q_{a_1,a_2} \circ p$.

Furthermore $[\![\underline{c}]\!]^{\mathfrak{N}}_{\Delta} = c \circ \Pi^{\Delta}_{\langle\rangle}$

$$= c \circ KI$$

$$= \lambda z.c(KIz)$$

$$= \lambda z.\underline{c}' .$$

Now we may proceed with the proof by induction,

$$[\![PQ]\!]^{\mathfrak{N}}_{\Delta} = ev \circ \langle f \circ [\![P]\!]^{\mathfrak{N}}_{\Delta}, [\![Q]\!]^{\mathfrak{N}}_{\Delta}\rangle$$

$$= ev_{bb} \circ \langle f \circ [\![\lambda z.P'[\ldots]]\!]^{\mathfrak{M}}, [\![\lambda z.Q'[\ldots]]\!]^{\mathfrak{M}}\rangle ,$$

by induction hypothesis,

$$= \lambda z.b((fP'[\ldots])(bQ'[\ldots]))$$

$$= \lambda z.b^{b}(fP'[\ldots])Q'[\ldots]$$

$$= \lambda z.fP'[\ldots]Q'[\ldots]$$

$$= \lambda z.(PQ)'[\ldots] .$$

For the last step in this inductive proof, assume that $y \notin |\Delta|$ and let

$\overset{*}{\ldots}$ stand for $x_n := zK(KI),\ldots,x_1 := zK^{\sim n}(KI)$,

$\ldots$ stand for $x_n := z(KI) ,\ldots,x_1 := zK^{\sim(n-1)}(KI)$.

Then

$$\begin{aligned}
[\![\lambda y.P(\vec{x},y)]\!]^{\mathfrak{N}}_{\Delta} &= g\circ\Lambda([\![P(\vec{x},y)]\!]^{\mathfrak{N}}_{\Delta;y})\\
&= g\circ\Lambda(\lambda z.P'[y := z(KI),.\overset{*}{.}.])\\
&= \lambda z.g(\lambda y.P'[y := z(KI),.\overset{*}{.}.][z := [z,y]])\\
&= \lambda z.g(\lambda y.P'[y := y,\ldots])\\
&= \lambda z.g(\lambda y.P')[\ldots]\\
&= \lambda z.(\lambda y.P)'[\ldots].
\end{aligned}$$

(iii) $$\begin{aligned}
(Kc_1)\cdot_{\mathfrak{N}}(Kc_2) &= [\![\underline{Kc_1}\ \underline{Kc_2}]\!]^{\mathfrak{N}}\\
&= \lambda z.(\underline{Kc_1}\ \underline{Kc_2})'\\
&= \lambda z.f(\underline{Kc_1}I)(\underline{Kc_2}I)\\
&= K(fc_1c_2).
\end{aligned}$$

□

Analogous to lemma 3.1.13. and proposition 3.1.14. we can embed the domain of $\mathfrak{N}$ into the domain of $\mathfrak{M}$, thereby arriving at the following situation.

3.2.4. Proposition. The model $\mathfrak{N}$ can be described (up to isomorphism) as

$|\mathfrak{N}| = \mathrm{Fix}(b) = \{c \in |\mathfrak{M}| \mid bc = c\}$;

$[\![P]\!]^{\mathfrak{N}} = [\![P']\!]^{\mathfrak{M}}$, for $P \in \Lambda^0(\mathfrak{N})$, where P' is as before except that now $\underline{c}' = \underline{c}$;

$c_1 \cdot_{\mathfrak{N}} c_2 = fc_1c_2$, for $c_1,c_2 \in |\mathfrak{N}|$.

Proof: Look at the retract (ρ,σ): $|\mathfrak{N}| \lhd |\mathfrak{M}(K(\mathfrak{M}))|$, defined by

$\rho(c) = c$, for $c \in |\mathfrak{N}|$,

$\sigma(a) = b\circ a$, for $a \in |\mathfrak{M}(K(\mathfrak{M}))|$.

Now take the image of this retract under the isomorphism

$\mathfrak{M}(K(\mathfrak{M})) \cong \mathfrak{M}$,

$Ka \leftrightarrow a$.

□

Using these facts about the relation between $\mathfrak{N}$ and $\mathfrak{M}$, we can give concrete descriptions of derived models of $\mathfrak{N}$, taken in the category $\mathcal{K}(\mathfrak{M})$. It will be shown that this situation is perfectly general in the sense that every derived model of $\mathfrak{N}$ can be represented inside $\mathcal{K}(\mathfrak{M})$.

3.2.5. Calculations.

(i) The retract $(\alpha,\beta)\colon \mathcal{K}(\mathfrak{M})(b,b) \lhd \mathcal{K}(\mathfrak{M})(T,b)$ is given by

$$\begin{aligned} \alpha(m) &= G \circ \Lambda(m \circ q_{Tb}) \\ &= g \circ \lambda xy.m(b(q[x,y])) \\ &= K(g(m \circ b)) \\ &= K(gm), \end{aligned}$$

$$\begin{aligned} \text{and} \quad \beta(x) &= ev \circ \langle F \circ x \circ ! \, , id_b \rangle \\ &= \lambda y.f(x(!y))(by) \\ &= f(xI) \circ b \\ &= f(xI) \quad \text{, since } f\colon b \to b^b. \end{aligned}$$

Under the isomorphism

$$\mathcal{K}(\mathfrak{M})(T,b) \cong Fix(b)$$

as used in proposition 3.2.4. we have

$$\alpha(m) = gm \quad \text{and} \quad \beta(x) = fx.$$

(ii) We suppose now that we have a retract $(r,s)\colon a \lhd b$ of the model $\mathfrak{N}$ inside the Karoubi-envelope of $\mathfrak{M}$,

$$\begin{tikzcd} b^b \arrow[r, shift left, "g"] & b \arrow[l, shift left, "f"] \arrow[d, shift right, "s"'] \\ & a \arrow[u, shift right, "r"'] \end{tikzcd} \quad .$$

It is easy to compute the following morphisms:

$$\bar{r} = \lambda x.r \circ x \circ s = r^s \; ,$$

$\bar{s} = \lambda x.s \circ x \circ r = s^r$, cf. the remarks after 3.1.1.,

$g' = \lambda x.s(g(r \circ x \circ s))$,

$f' = \lambda x.s \circ f(rx) \circ r$.

Now let $\delta = r \circ s$.

Claim: (BA) $\Longleftrightarrow$ $\delta^\delta \circ f \circ \delta \circ g \circ \delta^\delta = \delta^\delta$.

Proof: $f' \circ g' = \lambda x.s \circ f(r(s(g(r \circ x \circ s)))) \circ r$,

hence

$$\begin{aligned}
\text{(BA)} &\Longleftrightarrow f' \circ g' = id_a a \\
&\Longleftrightarrow \lambda x.s \circ f(\delta(g(r \circ x \circ s))) \circ r = \lambda x.a \circ x \circ a \\
&\overset{(*)}{\Longleftrightarrow} \lambda x.s \circ f(\delta(g(\delta \circ x \circ \delta))) \circ r = \lambda x.s \circ x \circ r \\
&\overset{(**)}{\Longleftrightarrow} \lambda x.\delta \circ f(\delta(g(\delta \circ x \circ \delta))) \circ \delta = \lambda x.\delta \circ x \circ \delta \\
&\Longleftrightarrow \delta^\delta \circ f \circ \delta \circ g \circ \delta^\delta = \delta^\delta.
\end{aligned}$$

(*) $\Rightarrow$ follows by applying both sides of the equation to $s \circ x \circ r$,

(*) $\Leftarrow$ follows by applying both sides of the equation to $r \circ x \circ s$,

(**) $\Rightarrow$ follows because $r \circ s = \delta$,

(**) $\Leftarrow$ follows because $s \circ \delta = s$ and $\delta \circ r = r$.

It is possible to prove this claim using theorem 3.1.9. and proposition 3.2.4., realizing that in this case $d^{\mathfrak{N}} = \alpha(\delta) = g\delta$ by (i).

□

From now on we will use the definition of the model $\mathfrak{N}$ as given in 3.2.4. The next proposition describes what the Karoubi-envelope of $\mathfrak{N}$ looks like.

3.2.6. Proposition. $\mathcal{K}(\mathfrak{N})$ is given by

(a) $|\mathcal{K}(\mathfrak{N})| = \{x \in |\mathfrak{N}| \mid fx \circ fx = fx \wedge g(fx) = x\}$,

(b) $\mathcal{K}(\mathfrak{N})(x,y) = \{m \in |\mathfrak{N}| \mid fy \circ fm \circ fx = fm \wedge g(fm) = m\}$.

Proof: First of all, let us compute $x \circ y$ in $\mathfrak{N}$:

$$\begin{aligned} x \circ_{\mathfrak{N}} y &= [\![\lambda z.x(yz)]\!]^{\mathfrak{N}} \\ &= g(\lambda z.fx(fy(bz))) \\ &= g(fx \circ fy \circ b) \\ &= g(fx \circ fy). \end{aligned}$$

Now, for $x,y,m \in |\mathfrak{N}|$:

(a) $$\begin{aligned} x \in |K(\mathfrak{N})| &\iff x \circ_{\mathfrak{N}} x = x \\ &\iff g(fx \circ fx) = x \\ &\iff fx \circ fx = fx \wedge g(fx) = x, \end{aligned}$$

(b) $$\begin{aligned} m \in K(\mathfrak{N})(x,y) &\iff y \circ_{\mathfrak{N}} m \circ_{\mathfrak{N}} x = m \\ &\iff g(fy \circ fm \circ fx) = m \\ &\iff fy \circ fm \circ fx = fm \wedge g(fm) = m. \end{aligned}$$

□

3.2.7. Corollary. There exists a full and faithful embedding $F\colon K(\mathfrak{N}) \to K(\mathfrak{M})$, given by

$$\begin{aligned} F(x) &= fx \ , \\ F(m\colon x \to y) &= fm\colon Fx \to Fy \ . \end{aligned}$$

Proof: Proposition 3.2.6. shows that F is welldefined, injective on objects and faithful.

F is indeed functorial:

Let $m\colon x \to y$, $n\colon y \to z$, then

$$\begin{aligned} F(n \circ_{\mathfrak{N}} m) &= F(g(fn \circ fm)) \\ &= f(g(fn \circ fm)) \\ &= fn \circ fm \\ &= F(n) \circ F(m) \end{aligned}$$

and

$$\begin{aligned} F(id\colon x \to x) &= F(x\colon x \to x) \\ &= fx\colon Fx \to Fx \\ &= id\colon Fx \to Fx. \end{aligned}$$

It remains to prove that $\mathcal{F}$ is full.
Let $m \in \mathcal{K}(\mathfrak{M})(\mathcal{F}x,\mathcal{F}y)$, then $m = fy \circ m \circ fx$,
hence

$$\begin{aligned} b^b m &= b \circ m \circ b \\ &= b \circ fy \circ m \circ fx \circ b \\ &= fy \circ m \circ fx \\ &= m. \end{aligned}$$

It follows that $f(gm) = b^b m = m$.
Therefore $gm \in \mathcal{K}(\mathfrak{N})(x,y)$ and $\mathcal{F}(gm) = m$.

□

Since we want to iterate the process of taking derived models, it would be convenient if we could reduce a double derivation to a single one. This transitivity of derived models follows easily, once a derived model of $\mathfrak{N}$ has been represented in $\mathcal{K}(\mathfrak{M})$.

3.2.8. Theorem. Let $\mathfrak{M}$ be a lambda algebra and $\mathfrak{N}$ the lambda algebra defined by the reflexive object b in $\mathcal{K}(\mathfrak{M})$,

$$\begin{tikzcd} b^b \arrow[r, shift left, "g"] & b \arrow[l, shift left, "f"] \end{tikzcd} .$$

Let $d \in |\mathfrak{N}|$ be a derivable element and $\mathfrak{N}(d)$ be the corresponding derived lambda algebra. Then $\mathfrak{N}(d)$ can be represented in $\mathcal{K}(\mathfrak{M})$ by

$$\begin{tikzcd} b^b \arrow[r, shift left, "g"] & b \arrow[l, shift left, "f"] \arrow[d, shift left, "fd"] \\ & fd \arrow[u, shift left, "fd"] \end{tikzcd} .$$

Proof: In $\mathcal{K}(\mathfrak{N})$ the derived lambda algebra $\mathfrak{N}(d)$ can be represented by the retract (d,d): $d \lhd I$ where $I = [\![I]\!]^{\mathfrak{N}} = g(\lambda x.bx) = gb$.
Applying the embedding functor $\mathcal{F}$ of corollary 3.2.7. to this retract we get the retract (fd,fd): $fd \lhd b$, given above.
We have to show that the element in $\mathfrak{N}$ that represents this retract is indeed d. But according to the calculations in 3.2.5.(i) this element is $\alpha(\delta) = \alpha(fd \circ fd)$

$$= g(fd \circ fd)$$

$$= d \qquad \text{, by proposition 3.2.6.(a), since } d \in |K(\mathfrak{N})|.$$

□

3.2.9. Corollary. Let $\mathfrak{N}$ be a derived lambda algebra of $\mathfrak{M}$ and $\mathfrak{N}'$ a derived lambda algebra of $\mathfrak{N}$. Then $\mathfrak{N}'$ is a derived lambda algebra of $\mathfrak{M}$. If d defines $\mathfrak{N}$ in $\mathfrak{M}$, that is $\mathfrak{N} = \mathfrak{M}(d)$, and d' defines $\mathfrak{N}'$ in $\mathfrak{N}$, that is $\mathfrak{N}' = \mathfrak{N}(d')$, then $d \circ d' \circ d$ defines $\mathfrak{N}'$ in $\mathfrak{M}$, that is $\mathfrak{N}' = \mathfrak{M}(d \circ d' \circ d)$. In other words $\mathfrak{M}(d)(d') = \mathfrak{M}(d \circ d' \circ d)$.

Proof: Consider the following diagram in $K(\mathfrak{M})$,

$$\begin{tikzcd}
1 \arrow[r, shift left, "1"] \arrow[d, shift right, "d^d"'] & I \arrow[l, shift left, "1"] \arrow[d, shift right, "d"'] & \mathfrak{M} \\
d^d \arrow[u, shift right, "d^d"'] \arrow[r, shift left, "g = d \circ d^d"] \arrow[d, shift right] & d \arrow[u, shift right, "d"'] \arrow[l, shift left, "f = d^d \circ d"] \arrow[d, shift right, "fd'"'] & \mathfrak{N} = \mathfrak{M}(d) \\
(fd')^{fd'} \arrow[u, shift right] \arrow[r, shift left] & fd' \arrow[u, shift right, "fd'"'] \arrow[l, shift left] & \mathfrak{N}' = \mathfrak{N}(d') \;;
\end{tikzcd}$$

$$fd' = (d^d \circ d)d' = d \circ dd' \circ d = d \circ d' \circ d, \text{ since}$$

$$d' \in |\mathfrak{N}| = \mathrm{Fix}(d).$$

□

This corollary describes iterations of derived models in general. There are particular cases in which this description is quite easy, even in the general case of an arbitrary categorical lambda algebra C.

3.2.10. Definition. Let $\mathfrak{M}$ be a lambda algebra, induced by a categorical lambda algebra C. The *canonical derived sequence* (with respect to $(G,F)\colon U^U \lhd U$) is defined by

$$U_0 = U \;,\quad G_0 = G \;,\quad F_0 = F \;,$$

$$U_{n+1} = U_n^{U_n} \;,\quad G_{n+1} = \bar{G}_n = G_n^{F_n} \;,\quad F_{n+1} = \bar{F}_n = F_n^{G_n} \,.$$

□

One should not confuse this canonical derived sequence with any of the sequences $(\varepsilon_n)_{n\geqslant 1}$ or $(1_n)_{n\geqslant 1}$ as defined in chapter 1.

3.2.11. Lemma. $(G_n,F_n)\colon U_{n+1}\lhd U_n$ represents the n-th canonical derived model of the model $(G,F)\colon U^U\lhd U$.

Proof: By induction on $n\in\omega$. The case for $n=0$ is clear. Now let $n = k+1$. By induction hypothesis $(G_k,F_k)\colon U_{k+1}\lhd U_k$ is the k-th canonical derived model. Construct the following diagram in $\mathcal{C}$,

$$\begin{tikzcd}
U_{k+1} \arrow[r, shift left, "G_k"] \arrow[d, shift right, "\bar{F}_k"'] & U_k \arrow[l, shift left, "F_k"] \arrow[d, shift right, "F_k"'] & \text{k-th derived model} \\
U_{k+2} \arrow[r, shift left, "G'"] \arrow[u, shift right, "\bar{G}_k"'] & U_{k+1} \arrow[l, shift left, "F'"] \arrow[u, shift right, "G_k"'] & \text{(k+1)-st derived model.}
\end{tikzcd}$$

Then $G' = F_k\circ G_k\circ \bar{G}_k = \bar{G}_k = G_{k+1}$,

$F' = \bar{F}_k\circ F_k\circ G_k = \bar{F}_k = F_{k+1}$.

Hence $(G_{k+1},F_{k+1})\colon U_{k+2}\lhd U_{k+1}$ is the (k+1)-st canonical derived model.

□

3.2.12. Remark. By taking $b = I$ and $f = g = 1$ in $\mathcal{K}(\mathfrak{M})$, definition 3.2.10. gives the canonical derived sequence as defined in Scott [1980A]. In this article Scott constructed, in the case of the lambda model $\mathbb{P}\omega$, the limit for this sequence, which turned out to be an extensional lambda model, $\mathbb{P}_\infty$, inside $\mathbb{P}\omega$. This construction will be analyzed in sections 3.3. and 3.4.

□

3.3. *Derived models and approximation.*

In this section we will consider derived models of categorical lambda models,

$$\begin{tikzcd}
{[X\to X]} \arrow[r, shift left, "G"] & X \arrow[l, shift left, "F"]
\end{tikzcd} ,$$

in the category CPO of complete partial orders with continuous maps (with respect to the Scott topology) as morphisms.

We remind the reader that

$$[X\to Y] = \{f\colon X\to Y \mid f \text{ continuous}\}$$

is an exponential Y^X in CPO and that F,G are continuous maps such that $F\circ G = \mathrm{id}_{[X\to X]}$. Hence $[X\to X]$ is also the set of representable functions on X.

In the remainder of this chapter we will make use of the concept of the *Böhmtree* of a lambda term, see Barendregt [1981], chapter 10. We will freely use notations introduced there.

Moreover in a CPO lambda model it is possible to interpret $\Lambda\bot$-terms, see Barendregt [1981], chapter 14.3., by stipulating that $[\![\bot]\!] = \bot$.

3.3.1. Definition. Let $\mathfrak{M} = (\mathrm{CPO},X,F,G)$ be a categorical lambda model.

(i) $\mathfrak{M}$ *has approximation*, if there exists a map $(\cdot)_\bullet\colon X\times\omega\to X$, such that for all $x,y\in X$ and $n,m\in\omega$

(a) $(\cdot)_n\colon X\to X$ is continuous,

(b) $n\leq m \Rightarrow (\cdot)_n\leq(\cdot)_m$,

(c) $x = \sup\{(x)_n \mid n\in\omega\}$,

(d) $(x)_0\, y \leq (x\bot)_0$,

(e) $(x)_{n+1}\, y \leq (x(y)_n)_n$.

(ii) $\mathfrak{M}$ is *strict* if $\mathfrak{M}\models \lambda x.\bot = \bot$.

□

From now on we will write "CPO lambda model (X,F,G)" instead of "categorical lambda model (CPO,X,F,G)".

3.3.2. Proposition. Let $\mathfrak{M} = (X,F,G)$ be a CPO lambda model. Then there exists a continuous $G'\colon [X\to X]\to X$ such that $\mathfrak{M}' = (X,F,G')$ is a strict CPO lambda model.

Proof: Let
$$G'(f) = \begin{cases} \bot & \text{, if } \forall x\in X(f(x)=\bot),\\ G(f) & \text{, otherwise.}\end{cases}$$

The fact is clearly proven once we have shown that

$$F(G'(f)) = f.$$

If $\exists x \in X(f(x) \neq \perp)$, then $G'(f) = G(f)$ and the result follows; if $\forall x \in X(f(x) = \perp)$, then

$$F(G'(f) \leqslant F(G(f)) = f \quad \text{, hence}$$
$$\forall x \in X\ (F(G'(f))(x) \leqslant f(x) = \perp).$$

Therefore $\qquad F(G'(f)) = f.$

□

3.3.3. Theorem. Let $\mathfrak{M} = (X,F,G)$ be a strict CPO lambda model, that has approximation $(\cdot)_\bullet : X \times \omega \to X$. Then

(a) The *approximation theorem holds in* $\mathfrak{M}$, that is

$$[\![P]\!]^{\mathfrak{M}} = \sup\{[\![P^{[k]}]\!]^{\mathfrak{M}} \mid k \in \omega\}.$$

(b) $P \underset{\sim}{\sqsubseteq} Q \Rightarrow \mathfrak{M} \models P \leqslant Q.$

Proof: For all $x \in X$ we have

$$\perp x = (\lambda x.\perp)x = \perp.$$

Now the proof of this theorem follows just as given in Barendregt [1981], 19.1.1.-19.1.11., for the special case of $\mathbb{P}\omega$. Notice that in this proof we do not need $((x)_n)_m = x_{\min(n,m)}$ and $(x)_0\, y = (x\perp)_0$, which hold for $\mathbb{P}\omega$, but only the weaker $((x)_n)_m \leqslant x_{\min(n,m)}$ and $(x)_0\, y \leqslant (x\perp)_0$, which follow easily from definition 3.3.1.

□

3.3.4. Corollary. Let $\mathfrak{M} = (X,F,G)$ be a strict CPO lambda model with approximation. Then $\mathfrak{M}$ satisfies $\mathfrak{B}$ (the theory of Böhmtree equality) and hence is sensible.

Proof: By proposition 3.3.3.(b).

□

3.3.5. Definition. Let $\mathfrak{M} = (X,F,G)$ be a CPO lambda model.

(i) $\mathfrak{M}$ is a *closure model* if $G \circ F \geqslant id_X$.

(ii) $\mathfrak{M}$ is an *extensional model* if $G \circ F = id_X$.

□

Note that $\mathfrak{M}$ is a closure model iff $\mathfrak{M} \models 1 \geqslant I$ and $\mathfrak{M}$ is an extensional model iff $\mathfrak{M} \models 1 = I$.

There are some other properties of $\mathbb{P}\omega$ and $\mathbb{D}_\infty$ that carry over to more general lambda models with approximation.

3.3.6. Proposition. Let $\mathfrak{M} = (X,F,G)$ be a strict closure model with approximation. Then for all $P,Q \in \Lambda\bot$

(i) $P \,{}^{\eta}\underset{\sim}{\sqsubseteq}\, Q \;\Rightarrow\; \mathfrak{M} \models P \leqslant Q$.

(ii) If, moreover, $\mathfrak{M}$ is extensional and nontrivial then

$$P \,{}^{\eta}\underset{\sim}{\sqsubseteq}{}^{\eta}\, Q \iff \mathfrak{M} \models P \leqslant Q.$$

In particular $\mathrm{Th}(\mathfrak{M}) = \mathcal{H}^*$.

Proof:

(i) Just as in Barendregt [1981], 19.1.12. - 19.1.14.

(ii) Just as in Barendregt [1981], 19.2.5. - 19.2.12.

□

Now that the constructions used for $\mathbb{P}\omega$ and $\mathbb{D}_\infty$ have been transferred to arbitrary CPO lambda models with approximation, it is interesting to see whether the process of derivation preserves this approximation structure. As we will see, in some cases in which the derivable element d has certain closure properties, this is true.

3.3.7. Definition. Let $\mathfrak{M} = (X,F,G)$ be a CPO lambda model.

(i) $a \in X$ is *closure-derivable* if $I \leqslant a = a \circ a \leqslant a^a$.

(ii) $a \in X$ is *strict* if $a\bot = \bot$.

□

3.3.8. Lemma. Let $\mathfrak{M} = (X,F,G)$ be a CPO lambda model.

(i) If $a \in X$ is closure-derivable, then a is derivable. Even stronger, we have $a^a \circ a = a \circ a^a = a^a$.

(ii) If a is (closure-) derivable, then so is a^a.

(iii) If $\mathfrak{M}$ is a strict model and $a \in X$ is strict, then so is a^a.

(iv) If $\mathfrak{M}$ is a closure model, then

$$I \leqslant a \wedge a \text{ is derivable } \Rightarrow a \text{ is closure-derivable.}$$

Proof:

(i) Let $a \in X$ be closure-derivable. Then

$$a^a \circ a \geqslant a^a = a^a \circ a^a \geqslant a \circ a^a \geqslant a^a = a^a \circ a^a \geqslant a^a \circ a.$$

We may conclude $a^a \circ a = a \circ a^a = a^a$.

(ii) If a is derivable, then so is a^a by lemma 3.1.18.
Now let a be closure-derivable. Then

(a) $a^a \geqslant a \geqslant I$, hence $a^a \geqslant I$,

(b) $a^a \circ a^a = a^a$, since $a \circ a = a$,

(c) $a \leqslant a^a$ and hence by monotonicity $a^a \leqslant (a^a)^{a^a}$.

By (a),(b) and (c) we conclude that a^a is closure-derivable.

(iii) Let $\mathfrak{M}$ be strict and $a \in X$ be strict. Then

$$\begin{aligned} a^a\bot = a \circ \bot \circ a = \lambda x.a(\bot(ax)) &= \lambda x.a\bot \\ &= \lambda x.\bot, \text{ since a is strict,} \\ &= \bot \text{ , since } \mathfrak{M} \text{ is strict.} \end{aligned}$$

Hence a^a is strict.

(iv) $\mathfrak{M}$ is a closure model, hence $\mathfrak{M} \models I \leqslant 1$.
Then in $\mathfrak{M}$

$$a^a = \lambda x.a \circ x \circ a \geqslant \lambda x.I \circ x \circ I = 1 \geqslant I,$$

and hence, using the fact that a is derivable,

$$a^a = a^a \circ a \circ a^a \geqslant I \circ a \circ I = \lambda x.ax = a.$$

□

After these preparations we can show that taking derived models preserves approximability in the case of a closure-derivable element. First we describe what a derived model in CPO looks like.

3.3.9. Proposition. Let $\mathfrak{M} = (X,F,G)$ be a CPO lambda model.
Let $a \in |\mathfrak{M}|$ be derivable. Then $(X(a),\leqslant,\bot_a)$ is a cpo, where

$$X(a) = \mathrm{Fix}(a) \text{ ,}$$
$$\bot_a = a\bot \text{ .}$$

Let moreover $F_a(x)(y) = a \cdot (x \cdot y)$ for $x,y \in X(a)$,
$G_a(f) = a \cdot G(x \mapsto f(a \cdot x))$ for $f \in [X(a) \to X(a)]$.
Then $\mathfrak{M}(a) = (X(a),F_a,G_a)$ is (isomorphic to) the derived model of $\mathfrak{M}$ induced by a.

Proof: Clearly $(X(a),\leqslant)$ is a partial order. Now for all $x \in X(a)$ we have $\bot_a = a\bot \leqslant ax = x$. Hence $\bot_a$ is the least element in X(a).

For $X(a)$ to be a cpo it suffices to show that for all directed $D \subseteq X(a)$ we have $\sup_X(D) \in X(a)$.
But

$$\begin{aligned} a \cdot \sup_X(D) &= \sup_X(a \cdot D) && \text{, by continuity,} \\ &= \sup_X(D) && \text{, since } D \subseteq X(a). \end{aligned}$$

Hence $\sup_X(D) \in X(a)$.
Now consider the following diagram in CPO

$$\begin{tikzcd}
{[X \to X]} \arrow[r, shift left, "G"] \arrow[d, shift right, "s_a^{r_a}"'] & X \arrow[l, shift left, "F"] \arrow[d, shift right, "s_a"'] \\
{[X(a) \to X(a)]} \arrow[r, shift left, "G_a"] \arrow[u, shift right, "r_a^{s_a}"'] & X(a) \arrow[l, shift left, "F_a"] \arrow[u, shift right, "r_a"']
\end{tikzcd} ,$$

where $s_a(x) = a \cdot x$ and $r_a(x) = x$.
An easy computation shows that the derivable element corresponding to this diagram is a.

□

3.3.10. Theorem. Let $\mathfrak{M} = (X,F,G)$ be a CPO lambda model with approximation $(\cdot)_\circ$. Let $a \in X$ be a closure-derivable element in $\mathfrak{M}$ and let $\mathfrak{M}(a)$ be the corresponding derived model. Then $\mathfrak{M}(a)$ also has approximation, defined by

$$[x]_n = a \cdot (x)_n, \text{ for all } x \in \mathfrak{M}(a).$$

Proof: We will verify conditions (a)-(e) in definition 3.3.1.(i).

(a) Since $[\cdot]_n = s_a \circ (\cdot)_n \circ r_a$ we have that $[\cdot]_n$ is a continuous mapping for all $n \in \omega$.

(b) Let $n \leqslant m$. Then $(x)_n \leqslant (x)_m$. By monotonicity it follows that $[x]_n = a(x)_n \leqslant a(x)_m = [x]_m$.

(c) Let $x \in X(a)$. Then
$x = a \cdot x = a \cdot \sup_X (x)_n = \sup_X a \cdot (x)_n = \sup_{X(a)} [x]_n$.

(d) Denoting application in $\mathfrak{M}(a)$ by $\cdot_a$ we get for $x,y \in X(a)$,

$$\begin{aligned} [x]_0 \cdot_a y &= a(a(x)_0 y) \\ &= a(a(x)_0 (ay)) && \text{, since } y \in X(a), \end{aligned}$$

$$
\begin{aligned}
&= (a^{a} \circ a)(x)_{0}\, y \\
&= a^{a}(x)_{0}\, y && \text{, by 3.3.8.(i),} \\
&= a((x)_{0}\, y) && \text{, since } y \in X(a)\text{,} \\
&\leqslant a(x\bot)_{0} \\
&\leqslant a(a(x(a\bot)))_{0} && \text{, since } a \geqslant I\text{,} \\
&= [x \bullet_{a} \bot_{a}]_{0}\,.
\end{aligned}
$$

(e) Similarly for all $x,y \in X(a)$ we have

$$
\begin{aligned}
[x]_{n+1} \bullet_{a} y &= a(a(x)_{n+1}\, y) \\
&= a((x)_{n+1}\, y) && \text{, as above,} \\
&\leqslant a(x(y)_{n})_{n} \\
&\leqslant a(a(x(a(y)_{n})))_{n} && \text{, since } a \geqslant I\text{,} \\
&= [x \bullet_{a} [y]_{n}]_{n}\,.
\end{aligned}
$$

□

Finally we want to take the notion of strictness into account, resulting in the following theorem about derivation and approximation.

3.3.11. Theorem. Let $\mathfrak{M}$ be a strict CPO lambda model with approximation. Let $a \in |\mathfrak{M}|$ be a strict closure-derivable element. Then the derived model $\mathfrak{M}(a)$ is a strict closure model with approximation.

Proof: Suppose that a is strict and closure-derivable and that $(\bullet)_{\bullet}$ defines an approximation in $\mathfrak{M}$. $\mathfrak{M}(a)$ has approximation $[\bullet]_{\bullet}$, by theorem 3.3.10.

Now

$$
\begin{aligned}
[\![\lambda x.\bot]\!]^{\mathfrak{M}(a)} &= a(\lambda x.a\bot) && \text{, by corollary 3.1.17.,} \\
&= a(\lambda x.\bot) && \text{, since a is strict,} \\
&= a\bot && \text{, since } \mathfrak{M} \text{ is strict,} \\
&= [\![\bot]\!]^{\mathfrak{M}(a)}.
\end{aligned}
$$

Therefore $\mathfrak{M}(a)$ is strict.

Also

$$\begin{aligned}
[\![1]\!]^{\mathfrak{M}(a)} &= a(\lambda x.a(\lambda y.a((ax)(ay)))) \\
&= a(a \circ a^{a} \circ a) \\
&= a \bullet a^{a} && \text{, by lemma 3.3.8.(i),} \\
&\geqslant a \bullet a && \text{, since a is closure-derivable,} \\
&= a(\lambda x.ax) \\
&= [\![I]\!]^{\mathfrak{M}(a)}.
\end{aligned}$$

Hence $\mathfrak{M}(a)$ is a closure model.

□

3.4. *The model* $\mathbb{P}_\infty$.

As an application of the method of taking derived models, we will now show how to find an extensional lambda model, defined entirely inside $\mathbb{P}\omega$. Since this model is a hybrid between $\mathbb{P}\omega$ and $\mathbb{D}_\infty$ it will be denoted by $\mathbb{P}_\infty$. The original construction of $\mathbb{P}_\infty$ appeared in print in Scott [1974]. The first analysis of the theory of $\mathbb{P}_\infty$ was given by Laarhoven [1975].

In this section we will use the theory of derived models to show that the theory of $\mathbb{P}_\infty$ is in fact $\mathcal{H}^*$, hence equals the theory of any of the $\mathbb{D}_\infty$-models of Scott [1972]. $\mathbb{P}_\infty$ is defined as the limit of a sequence $(\mathbb{P}_n)_{n\in\omega}$ of models, which will be shown to have the same theory as $\mathbb{P}\omega$. This is no more strange than two converging sequences $(x_n)_{n\in\omega}$ and $(y_n)_{n\in\omega}$ of real numbers with $x_n \neq y_n$ for all $n \in \omega$ and yet $\lim_{n\to\infty} x_n = \lim_{n\to\infty} y_n$.

Since the construction will not only be applicable to $\mathbb{P}\omega$, but also to other CPO lambda models, we will start with such an arbitrary $\mathfrak{M} = (X,F,G)$ in CPO and try to construct the model $\mathfrak{M}_\infty$.

3.4.1. Definition. Let $(D_n)_{n\in\omega}$ be the sequence of closed lambda terms, defined by induction on $n \in \omega$ as follows

$$D_0 = I \quad ,$$

$$D_{n+1} = D_n^{D_n} = \lambda f.D_n \circ f \circ D_n \,.$$

□

This is the notation, as given in Scott [1980A], for the canonical derived sequence in $K(\mathfrak{M})$, for any lambda algebra $\mathfrak{M}$, with respect to $(g,f)\colon b^b \lhd b$, where $b = I = D_0$ and $g = f = b^b = I^I = 1 = D_1$. Then it follows for arbitrary $n \in \omega$, that $b_n = D_n$ and $g_n = f_n = D_{n+1}$.

3.4.2. Observation. By lemma 3.2.11. we know that the n-th element of the canonical derived sequence represents the n-th canonical derived model. We may draw a diagram in $K(\mathfrak{M}^0(\lambda))$:

$$\begin{tikzcd}
1 = D_1 \arrow[r, shift left, "D_1"] \arrow[d, shift right, "D_2"'] & D_0 = I \arrow[l, shift left, "D_1"] \arrow[d, shift right, "D_1"'] \\
D_2 \arrow[u, shift right, "D_2"'] \arrow[r, shift left, "D_2"] \arrow[d, shift right] & D_1 \arrow[u, shift right, "D_1"'] \arrow[l, shift left, "D_2"] \arrow[d, shift right] \\
 \arrow[u, shift right] & \arrow[u, shift right]
\end{tikzcd}$$

$$\vdots \qquad .$$

From this diagram we read off that

$$D_n \in |K(\mathfrak{M}^0(\lambda))| \ ,$$

$$D_{n+1} \in K(\mathfrak{M}^0(\lambda))\,(D_n, D_{n+1}) \quad \text{and}$$

$$D_{n+1} \in K(\mathfrak{M}^0(\lambda))\,(D_{n+1}, D_n) \ .$$

Hence $D_n \circ D_n = D_n$, $D_{n+1} \circ D_n = D_{n+1}$ and $D_n \circ D_{n+1} = D_{n+1}$.

These equations can also be proved directly by induction on n.

□

For the remainder of this section let us fix a CPO lambda model $\mathfrak{M}$. We write $d_n = [\![D_n]\!]^{\mathfrak{M}}$. The n-th canonical derived model will be denoted by $\mathfrak{M}_n = \mathfrak{M}(d_n)$. We are going to define a kind of limit model for this sequence. In order to do this we need to know that the sequence $(d_n)_{n\in\omega}$ is directed.

3.4.3. Lemma. Let $\mathfrak{M}$ be a closure model. Then for all $n \in \omega$ we have $d_n \leqslant d_{n+1}$.

Proof: By induction on $n \in \omega$.
For $n = 0$ we have $d_0 = I \leqslant 1 = d_1$, since $\mathfrak{M}$ is a closure model.
Now assume by induction hypothesis $d_n \leqslant d_{n+1}$.
Then

$$\begin{aligned} d_{n+2} &= \lambda x.d_{n+1} \circ x \circ d_{n+1} \\ &\geqslant \lambda x.d_n \circ x \circ d_n \\ &= d_{n+1}. \end{aligned}$$

□

Because of this lemma we are able to consider the limit (supremum) of the sequence $(d_n)_{n\in\omega}$. For any closure model $\mathfrak{M}$ let $d_\infty = \sup\{d_n \mid n \in \omega\}$. Next we want to show that d_∞ not only induces a derived model, but even an extensional one.

3.4.4. Lemma. Let $\mathfrak{M}$ be a (strict) closure model.
(i) $\forall n \in \omega$ (d_n is (strict) closure-derivable).
(ii) d_∞ is (strict) closure-derivable and strongly extensional.

Proof:
(i) By lemma 3.3.8.(ii),(iii).
(ii) Clearly $I \leqslant d_\infty$, since $I \leqslant d_n$ for all n.
Furthermore

$$\begin{aligned} d_\infty \circ d_\infty &= \sup\{d_n \mid n \in \omega\} \circ \sup\{d_m \mid m \in \omega\} \\ &= \sup\{d_n \circ d_m \mid n,m \in \omega\}, \text{ by continuity,} \\ &= \sup\{d_n \mid n \in \omega\} \\ &= d_\infty. \end{aligned}$$

And also

$$\begin{aligned} d_\infty^{d_\infty} &= \lambda x.d_\infty \circ x \circ d_\infty \\ &= \sup\{\lambda x.d_n \circ x \circ d_n \mid n \in \omega\}, \text{ by continuity,} \\ &= \sup\{d_{n+1} \mid n \in \omega\} \\ &= d_\infty. \end{aligned}$$

□

Now we may define $\mathfrak{M}_\infty = \mathfrak{M}(d_\infty)$. According to lemma 3.1.23. we have

$$|\mathfrak{M}_\infty| = \mathrm{Fix}(d_\infty)\ ,$$
$$a \cdot_{d_\infty} b = ab\ ,\ \text{ for } a,b \in |\mathfrak{M}_\infty|,$$
$$[\![P]\!]^{d_\infty} = [\![P \restriction d_\infty]\!]^{\mathfrak{M}}.$$

Moreover $\mathfrak{M}_\infty$ is an extensional lambda model by corollary 3.1.21.

If $\mathfrak{M}$ is a strict closure model with approximation, then we can say even more.

3.4.5. Proposition. Let $\mathfrak{M}$ be a strict closure model with approximation.

(i) $\forall n$ ($\mathfrak{M}_n$ is a strict closure model with approximation).

(ii) $\mathfrak{M}_\infty$ is an extensional model with approximation.

Proof:

(i) By theorem 3.3.11. and lemma 3.4.4.(i).

(ii) By theorem 3.3.11., lemma 3.4.4.(ii) and corollary 3.1.21.

□

Let us remark here that if $\mathfrak{M}$ is nontrivial, then so is $\mathfrak{M}_\infty$. For, $\bot = d_\infty\bot \in |\mathfrak{M}_\infty|$ and if $c \neq \bot$, then $c = Ic \leqslant d_\infty c \in |\mathfrak{M}_\infty|$. Hence $|\mathfrak{M}_\infty|$ contains at least the two different elements $\bot$ and $d_\infty c$.

3.4.6. Corollary. Let $\mathfrak{M}$ be a nontrivial strict closure model with approximation.

(i) For all $n \in \omega$, $P,Q \in \Lambda\bot$ we have

$$P \;{}^{n}\underset{\sim}{\sqsubseteq}\; Q \;\Rightarrow\; \mathfrak{M}_n \models P \leqslant Q.$$

(ii) For all $P,Q \in \Lambda\bot$

$$P \;{}^{n}\underset{\sim}{\sqsubseteq}{}^{n}\; Q \;\Longleftrightarrow\; \mathfrak{M}_\infty \models P \leqslant Q.$$

In particular $\mathrm{Th}(\mathfrak{M}_\infty) = \mathcal{H}^* = \mathrm{Th}(\mathbb{D}_\infty)$.

Proof: By propositions 3.3.6. and 3.4.5.

□

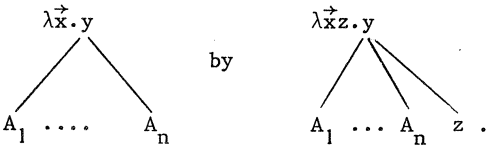
In certain cases we can sharpen the result of corollary 3.4.6.(i) by getting an equivalence instead of an implication. In fact it is the case that if this equivalence holds for $\mathfrak{M}$ then it also holds for any $\mathfrak{M}_n$. In particular all of this is true for $\mathbb{P}\omega$.

First of all we need more information about the translation $M \mapsto M^1$ corresponding to taking the canonical derived model.

3.4.7. Proposition. Let $M \in \Lambda\bot$.

(i) If M is unsolvable, then so is M^1.

(ii) $BT(M^1)$ is constructed from $BT(M)$ by replacing *every* node of the form



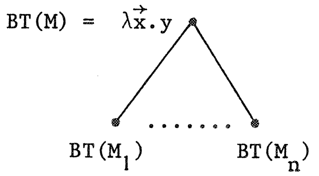
Proof:

(i) Semantic proof: If M is unsolvable, then by corollary 3.4.6.(i) we have $(\mathbb{P}\omega)_1 \models M = \bot$. Hence $\mathbb{P}\omega \models M^1 = \bot^1 = \bot$. By the characterization theorem for $\mathbb{P}\omega$ we have M^1 is unsolvable.

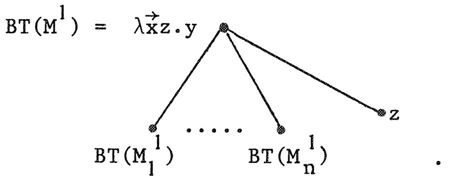
Syntactic proof: Clearly $M =_{\beta\eta} M^1$. If M^1 is β-solvable, then M is $\beta\eta$-solvable, hence β-solvable (see Barendregt [1981], proposition 15.1.7.).

(ii) If M is unsolvable, then so is M^1 by (i). Hence $BT(M) = \bot = BT(M^1)$. Now assume M is solvable, say $M = \lambda\vec{x}.yM_1\ldots M_n$. Then clearly $M^1 = \lambda\vec{x}z.yM_1^1\ldots M_n^1 z$.

Hence



and



.

The proposition follows.

□

Because of this proposition it is natural to introduce an operation $A \mapsto A^1$ on Böhm like trees such that $BT(M^1) = BT(M)^1$ as follows.

3.4.8. Definition. Let $A \in \mathfrak{B}$. Define $A^1 \in \mathfrak{B}$ by making at every node $\alpha \in A$, such that $A(\alpha) \neq \perp$, the smallest possible, nontrivial η-expansion.

□

It is clearly possible to define A^1 as $(A;X)$ for some suitable X, but since there is no gain in clearity doing so, we won't bother. For the notation $(A;X)$, see Barendregt [1981], definition 10.2.10.

Similarly the following lemma can be proved formally by making extensive use of the properties of $(A;X)$. But intuitively this lemma can be easily understood by drawing the relevant Böhm trees.

3.4.9. Lemma. Let $A,B \in \mathfrak{B}$.

(i) $A \leqslant_\eta A^1$.

(ii) $A^1 \leqslant_\eta B^1 \Rightarrow A \leqslant_\eta B$.

(iii) $A^1 \subseteq B^1 \Rightarrow A \subseteq B$.

(iv) If $C_1 \in \mathfrak{B}$ such that $A^1 \leqslant_\eta C_1 \subseteq B^1$, then there exists a $C \in \mathfrak{B}$ such that $C_1 = C^1$.

Proof:

(i) Clear by definition 3.4.8.

(ii) Suppose $A^1 \leqslant_\eta B^1$. Then, by (i), $A,B \leqslant_\eta B^1$.
Therefore $\infty\eta(B^1) \leqslant_\eta A,B$. Clearly also $|A| \subseteq |B|$.
We conclude $A \leqslant_\eta B$.

(iii) Suppose $A(\alpha) = \langle \lambda\vec{x}.y,k \rangle$.
Then $A^1(\alpha) = \langle \lambda\vec{x}z^\alpha.y,k+1 \rangle = B^1(\alpha)$ and $A^1(\alpha * \langle k \rangle) = \langle z^\alpha,0 \rangle = B^1(\alpha * \langle k \rangle)$
Therefore $B(\alpha) = \langle \lambda\vec{x}.y,k \rangle = A(\alpha)$.

(iv) Let

$$C(\alpha) = \begin{array}{ll} \uparrow & \text{, if } C_1(\alpha)\uparrow\text{,} \\ B(\alpha) & \text{, otherwise .} \end{array}$$

Note that $C_1(\alpha) = \perp$ implies $\alpha \in |B|$, since otherwise there exists a rightmost branch in C_1, ending in $\perp$, which contradicts $A^1 \leqslant_\eta C_1$.

From this observation it follows easily that $C_1 = C^1$.

□

3.4.10. Corollary. For all $P,Q \in \Lambda\perp$

$$P^1\ {}^{\eta}\underset{\sim}{\sqsubseteq}\ Q^1 \Rightarrow P\ {}^{\eta}\underset{\sim}{\sqsubseteq}\ Q.$$

Proof: Suppose $P^1\ {}^{\eta}\underset{\sim}{\sqsubseteq}\ Q^1$. Then there exists $C_1 \in \mathfrak{B}$ such that $BT(P)^1 \leqslant_\eta C_1 \subseteq BT(Q)^1$. By lemma 3.4.9.(iv) we may pick a $C \in \mathfrak{B}$ such that $C_1 = C^1$. Applying lemma 3.4.9.(ii),(iii) this gives

$$BT(P) \leqslant_\eta C \subseteq BT(Q).$$

Hence $P\ {}^{\eta}\underset{\sim}{\sqsubseteq}\ Q$.

□

With this result we can give the promised sharpening of corollary 3.4.6.(i).

3.4.11. Corollary. Let $\mathfrak{M}$ be a nontrivial strict closure model with approximation, such that for $P,Q \in \Lambda\perp$

$$\mathfrak{M} \models P \leqslant Q \iff P\ {}^{\eta}\underset{\sim}{\sqsubseteq}\ Q.$$

Then for all $n \in \omega, P,Q \in \Lambda\perp$ we have

$$\mathfrak{M}_n \models P \leqslant Q \iff \mathfrak{M} \models P \leqslant Q.$$

Proof:

($\Leftarrow$) this is corollary 3.4.6.(i).

($\Rightarrow$) It is clearly enough to show this for $n = 1$, since $\mathfrak{M}_{n+1} = (\mathfrak{M}_n)_1$. But

$$\begin{aligned}\mathfrak{M}_1 \models P \leqslant Q &\Rightarrow \mathfrak{M} \models P^1 \leqslant Q^1 \\ &\Rightarrow P^1\ {}^{\eta}\underset{\sim}{\sqsubseteq}\ Q^1 \\ &\Rightarrow P\ {}^{\eta}\underset{\sim}{\sqsubseteq}\ Q \quad \text{, by corollary 3.4.10.,} \\ &\Rightarrow \mathfrak{M} \models P \leqslant Q.\end{aligned}$$

□

Let me now state the results for $\mathbb{P}\omega$ in a separate theorem.

3.4.12. Theorem. Let $\mathbb{P}_n = (\mathbb{P}\omega)_n$, $\mathbb{P}_\infty = (\mathbb{P}\omega)_\infty$.

(i) For all $n \in \omega$, the model $\mathbb{P}_n$ is a strict closure model with approximation and

$$\mathbb{P}_n \models P \leqslant Q \iff P \;{}^{\eta}\!\underset{\sim}{\sqsubset}\; Q.$$

In particular $\mathrm{Th}(\mathbb{P}_n) = \mathrm{Th}(\mathbb{P}\omega)$.

(ii) $\mathbb{P}_\infty$ is an extensional lambda model with approximation and

$$\mathbb{P}_\infty \models P \leqslant Q \iff P \;{}^{\eta}\!\underset{\sim}{\sqsubset}{}^{\eta}\; Q.$$

In particular $\mathrm{Th}(\mathbb{P}_\infty) = \mathrm{Th}(\mathbb{D}_\infty)$.

Proof: $\mathbb{P}\omega$ has approximation given by

$$(x)_n = x \cap \{0,\dots,n\} \text{ for } x \subseteq \omega,\ n \in \omega.$$

We refer the reader to Barendregt [1981], 18.2.14.
Clearly $\mathbb{P}\omega$ is strict and nontrivial. Moreover, by example 1.5.5., $\mathbb{P}\omega$ is a closure model.

(i) We have

$$\mathbb{P}\omega \models P \leqslant Q \iff P \;{}^{\eta}\!\underset{\sim}{\sqsubset}\; Q$$

by a theorem of Hyland, see Barendregt [1981], 19.1.9.
The result follows by corollary 3.4.11.

(ii) Apply corollary 3.4.6.(ii).

□

Note that the sequence

$$\mathrm{BT}(P),\ \mathrm{BT}(P^1),\ \mathrm{BT}((P^1)^1),\dots.$$

approximates the *Nakajima tree* NT(P), as introduced in Nakajima [1975]. Theorem 3.4.12.(ii) fits in nicely with the following result of Nakajima (cf. Barendregt [1981], 19.4.4.):

$$\mathrm{NT}(P) \subseteq \mathrm{NT}(Q) \iff \mathbb{D}_\infty \models P \leqslant Q.$$

CHAPTER 4

THE HYPERGRAPHMODEL

In Sanchis [1979] a model for combinatory logic is introduced, which is a Π^1_1-variant of the wellknown graphmodel $\mathbb{P}\omega$ for lambda calculus, as defined in example 1.4.31. It was unclear in howfar this socalled hypergraphmodel satisfied the laws of lambda calculus. In this chapter it will be shown that the hypergraphmodel can *not* be expanded to a lambda model. The question whether it can be expanded to a lambda algebra remains open.

4.1. *Fundamental definitions and properties.*

It is the purpose of this first section to introduce the hypergraphmodel for combinatory logic of Sanchis [1979] and to give the necessary facts about it and the ordinary graphmodel $\mathbb{P}\omega$ as a preparation for the next section in which it will be proved that the Meyer-Scott axiom fails for the hypergraphmodel. We will stay quite close to the original presentation of Sanchis, introducing some more notation where convenient.

4.1.1. Notations.

(i) $\langle \cdot,\cdot \rangle : \omega^2 \to \omega$ is a bijective coding of pairs of natural numbers onto natural numbers.

$n \mapsto e_n : \omega \to \{e \subseteq \omega \mid e \text{ is finite}\}$ is a bijective coding of finite subsets of the natural numbers by natural numbers.

(ii) For $n,m,p \in \omega$ let $\langle n,m,p \rangle = \langle n, \langle m,p \rangle \rangle$.

This gives a bijective coding of ω^3 onto ω. In any of the standard ways we may define an injective coding of the set of all finite sequences of natural numbers into ω.

Let $(a(0),\ldots,a(k-1))$ be the code of the sequence $a: k \to \omega$.

If $f: \omega \to \omega$ and $p \in \omega$ then $\bar{f}(p) = (f(0),\ldots,f(p-1))$.

(iii) $\mathrm{Seq} = \{\bar{f}(p) \in \omega \mid f\colon \omega \to \omega \wedge p \in \omega\}$ is the set of all sequence numbers.

$\alpha, \beta, \gamma \ldots$ are variables that range over Seq.

$\alpha \prec \beta$ means that α codes a sequence that is an initial segment of (or equal to) the sequence β codes.

□

Unless stated otherwise, in this chapter, $n,m,p,\ldots$ range over ω; $f,g,h,\ldots$ range over ω^ω; $A,B,C,\ldots$, $X,Y,Z,\ldots$ range over $P\omega$ and $\alpha,\beta,\gamma,\ldots$ range over Seq.

4.1.2. Definition. Let $A \subseteq \omega$ and $\alpha \in \mathrm{Seq}$. Then the *α-slice of* A is defined by

$$A_\alpha = \{\langle n,m\rangle \mid \langle \alpha,n,m\rangle \in A\}.$$

□

The operation $A \mapsto (A_\alpha)_{\alpha \in \mathrm{Seq}}$ decomposes any set of natural numbers into a tree of such sets. This will become important in the definition of application in the hypergraphmodel. To have all tools at hand we will repeat here also the definition of application in $\mathbb{P}\omega$.

4.1.3. Definition. Let the binary operations $\bullet : P\omega^2 \to P\omega$ and $! : P\omega^2 \to P\omega$ be defined by

$$A \bullet B = \{m \mid \exists e_n \subseteq B(\langle n,m\rangle \in A)\},$$

$$A!B = \{m \mid \forall f \exists p \exists e_n \subseteq B(\langle \bar{f}(p),n,m\rangle \in A)\}.$$

□

Now we have two applicative structures,

$\mathbb{P}\omega = (P\omega, \bullet)$, the graphmodel, and

$\mathbb{H}\omega = (P\omega, !)$, the hypergraphmodel.

Both structures can be expanded to combinatory algebras.

4.1.4. Theorem.

(i) $\mathbb{P}\omega$ is combinatory complete. Indeed, it can be expanded to a lambda model.

(ii) $\mathbb{H}\omega$ is combinatory complete.

Proof:

(i) This follows from the fact that $\mathbb{P}\omega$ can easily be represented in the category CPO in such a way that a continuous graphfunction can be defined. This was done in example 1.4.31.

(ii) This is a result of Sanchis [1979], §2 theorem 4.

□

Reading this theorem, a natural question to ask is whether $\mathbb{H}\omega$ can be expanded to a lambda model. An approach as in 4.1.4.(i) seems quite ineffective, since the hypergraphapplication $! : P\omega^2 \to P\omega$ is hopelessly noncontinuous with respect to the Scott topology on $(P\omega, \subseteq)$. In fact it will turn out that $\mathbb{H}\omega$ cannot be expanded to a lambda model.

4.1.5. Notation. We will make use of the following notation, introduced by Sanchis:

$$(AB) = A \cdot B \ ,$$

$$[AB] = A!B \ .$$

□

When using this notation we need to be very precise about writing brackets.

4.1.6. Remark. In order to get a better understanding how hypergraphapplication works, we may realize that

$$m \in [AB] \iff \forall f \exists p (m \in (A_{\bar{f}(p)} B))$$

$\iff$ There exists a bar $\mathfrak{B}$ in the tree of sequence numbers such that $\forall \alpha \in \mathfrak{B}(m \in (A_\alpha B))$.

This means that A codes a whole tree of sets $(A_\alpha)_{\alpha \in Seq}$, the tree of its α-slices. This tree can be applied pointwise in the ordinary graphapplication-sense, resulting in another tree of sets. This tree then determines the result by looking at its bars.

□

The remainder of this section gives some more preparations for the next section.

4.1.7. Lemma.

(i) Both $\cdot$ and $!$ are monotonic in both their arguments.

(ii) If $\{A_i \mid i \in I\}$ is any set of elements of $P\omega$ then

$$(\bigcup_{i\in I} A_i)\cdot B \;=\; \bigcup_{i\in I}(A_i\cdot B).$$

Proof:

(i) Immediate from definition 4.1.3.

(ii) For all $m\in\omega$

$$\begin{aligned} m\in(\bigcup_{i\in I}A_i)\cdot B &\iff \exists e_n\subseteq B(\langle n,m\rangle\in\bigcup_{i\in I}A_i)\\ &\iff \exists i\in I\ \exists e_n\subseteq B(\langle n,m\rangle\in A_i)\\ &\iff \exists i\in I\ m\in(A_i\cdot B)\\ &\iff m\in\bigcup_{i\in I}(A_i\cdot B). \end{aligned}$$

□

Notice that the fact that graphapplication preserves arbitrary unions in its first argument is a kind of supercontinuity in that argument, also called additivity. Indeed, continuity states that the union preserving property holds for directed sets $\{A_i \mid i\in I\}$.

Let me remind the reader of the following property of $1=[\![\lambda xy.xy]\!]^{P\omega}$. For all $A\in P\omega$

$$\langle n,m\rangle\in(1A) \iff \exists e_k\subseteq e_n(\langle k,m\rangle\in A).$$

This follows immediately from example 1.5.5., since $(1A) = G(F(A))$.

In the hypergraphcase we have the following analogous definition.

4.1.8. Definition.

(i) The operation $^{-}: P\omega\to P\omega$ is defined by

$$\bar{A} = \{\langle\alpha,n,m\rangle \mid \exists\beta\prec\alpha\ \exists e_k\subseteq e_n(\langle\beta,k,m\rangle\in A)\}.$$

(ii) The binary operation $\bar{\cap}: P\omega^2\to P\omega$ is defined by

$$\bar{\cap}(A,B) = \bar{A}\cap\bar{B}.$$

□

Notice that $^{-}$ is a kind of saturation operation since

$$\bar{A}_\alpha = \bigcup\{(1A_\beta)\mid \beta\prec\alpha\}.$$

Also note that in general $\bar{\cap}(A,B)\neq\overline{A\cap B}$.

4.1.9. Proposition. Let $A,B \subseteq \omega$.

(i) $\forall X \subseteq \omega \quad [\bar{A}X] = [AX]$.

(ii) $\forall X \subseteq \omega \quad [\bar{\cap}(A,B)X] = [AX] \cap [BX]$.

Proof:

(i) Let $X \subseteq \omega$, $m \in \omega$. Then

$$\begin{aligned} m \in [\bar{A}X] &\iff \forall f \exists p \exists e_n \subseteq X (\langle \bar{f}(p), n, m \rangle \in \bar{A}) \\ &\iff \forall f \exists p \exists e_n \subseteq X \ \exists \beta \prec \bar{f}(p) \exists e_k \subseteq e_n (\langle \beta, k, m \rangle \in A) \\ &\iff \forall f \exists q \exists e_k \subseteq X (\langle \bar{f}(q), k, m \rangle \in A) \\ &\iff m \in [AX]. \end{aligned}$$

(ii) Let $X \subseteq \omega$, $m \in \omega$. Then $m \in [AX] \cap [BX]$

$$\begin{aligned} &\iff \forall f \exists p_1, p_2 \ \exists e_{n_1}, e_{n_2} \subseteq X (\langle \bar{f}(p_1), n_1, m \rangle \in A \wedge \langle \bar{f}(p_2), n_2, m \rangle \in B) \\ &\overset{(*)}{\iff} \forall f \exists p \exists e_n \subseteq X (\exists p_1 \leqslant p \ \exists e_{n_1} \subseteq e_n (\langle \bar{f}(p_1), n_1, m \rangle \in A) \\ &\qquad\qquad \wedge \ \exists p_2 \leqslant p \ \exists e_{n_2} \subseteq e_n (\langle \bar{f}(p_2), n_2, m \rangle \in B)) \\ &\iff \forall f \exists p \exists e_n \subseteq X (\langle \bar{f}(p), n, m \rangle \in \bar{A} \cap \bar{B}) \\ &\iff m \in [\bar{\cap}(A,B)X]. \end{aligned}$$

(*) Hint: $p = \max(p_1, p_2)$ and $e_n = e_{n_1} \cup e_{n_2}$. □

4.2. *$\mathbb{H}\omega$ cannot be expanded to a lambda model.*

In order to refute that $\mathbb{H}\omega$ can be expanded to a lambda model we will make use of the Meyer-Scott axiomatization of lambda models as given in section 1.4.

Let us recall this axiomatization:

$\mathbb{H}\omega$ can be expanded to a lambda model $\iff$
There exists an element $\varepsilon \in P\omega$ such that

$$(MS_\varepsilon) \begin{cases} (MS_\varepsilon 1) & \forall A, X \subseteq \omega \quad [[\varepsilon A]X] = [AX] \text{ and} \\ (MS_\varepsilon 2) & \text{For all } A, B \subseteq \omega \\ & \forall X([AX] = [BX]) \Rightarrow [\varepsilon A] = [\varepsilon B]. \end{cases}$$

Hence in order to show that the hypergraphmodel cannot be expanded to a lambda model it is sufficient to derive a contradiction from (MS_ε).

The fact that $(MS_\varepsilon 2)$ holds implies the corresponding statement for inclusion.

4.2.1. Lemma. If (MS_ε) holds then also

$(MS_\varepsilon 3)$ For all $A,B \subseteq \omega$
$\forall X([AX] \subseteq [BX]) \Rightarrow [\varepsilon A] \subseteq [\varepsilon B]$.

Proof:

$$\begin{aligned}
\forall X([AX] \subseteq [BX]) &\Longleftrightarrow \forall X([AX] \cap [BX] = [AX]) \\
&\Longleftrightarrow \forall X([\bar{\cap}(A,B)X] = [AX]),\ \text{by proposition 4.1.9.(ii)}, \\
&\Longleftrightarrow [\varepsilon \bar{\cap}(A,B)] = [\varepsilon A],\ \text{by } (MS_\varepsilon), \\
&\Rightarrow [\varepsilon A] \subseteq [\varepsilon \bar{B}],\ \text{by lemma 4.1.7.(i)}, \\
&\Longleftrightarrow [\varepsilon A] \subseteq [\varepsilon B],\ \text{by proposition 4.1.9.(i) and } (MS_\varepsilon 2).
\end{aligned}$$

(In fact the arrow ⇒ in this proof is also an equivalence, since $[\varepsilon A] \subseteq [\varepsilon B]$ implies $\forall X([AX] \subseteq [BX])$ by monotonicity and $(MS_\varepsilon 1)$.) □

Now our purpose is to construct for a given set C two other sets A_C and B_C that are quite similar in their applicative behaviour. In fact A_C and B_C will differ on only one argument, viz. C. Although A_C and B_C are so very similar, $[\varepsilon A_C]$ has to be different from $[\varepsilon B_C]$, if (MS_ε) holds. The proof that (MS_ε) is contradictory will depend heavily on this discriminating power of ε.

4.2.2. Definition. Let $C \subseteq \omega$, $p \in \omega$.

Define functions f_C: $P\omega \to P\omega$ and $g_{p,C}$: $P\omega \to P\omega$ as follows:

$$f_C(X) = \begin{cases} \emptyset, & \text{if } X \subseteq C, \\ \omega, & \text{if } X \not\subseteq C. \end{cases}$$

$$g_{p,C}(X) = \begin{cases} \emptyset, & \text{if } X \not\supseteq C \cap e_p, \\ \omega, & \text{if } X \supseteq C \cap e_p. \end{cases}$$

□

4.2.3. Lemma. For all $C \subseteq \omega$, $p \in \omega$ we have that f_C and $g_{p,C}$ are continuous functions.

Proof: In any complete partial order $(X,\leqslant,\bot)$ it is the case that $\{x \mid x \not\leqslant z\}$ is open. Hence f_C is continuous.
In $(P\omega,\subseteq)$ the compact elements are the finite sets. Hence $\{X \mid X \supseteq C \cap e_p\}$ is open. Therefore $g_{p,C}$ is continuous. □

4.2.4. Definition. Let $C \subseteq \omega$.
Define $A_C = \mathrm{Seq} \times \mathrm{graph}(f_C) \subseteq \omega$

$$\text{and } B_C = A_C \cup \bigcup_{p \in \omega} \{(p)\} \times \mathrm{graph}(g_{p,C})$$

$$= A_C \cup \{\langle (p),n,m\rangle \mid \langle n,m\rangle \in \mathrm{graph}(g_{p,C})\}.$$

(Here graph is the graph-operator in $\mathbb{P}\omega$; recall furthermore that (p) is the code for the single term sequence p.) □

4.2.5. Proposition. Let $C \subseteq \omega$.

(a) For all $X \subseteq \omega$ we have

$$[A_C X] = \begin{cases} \emptyset, & \text{if } X \subseteq C, \\ \omega, & \text{if } X \not\subseteq C. \end{cases}$$

(b) For all $X \subseteq \omega$ we have

$$[B_C X] = \begin{cases} \emptyset, & \text{if } X \subsetneq C, \\ \omega, & \text{otherwise.} \end{cases}$$

Proof:

(a) Since for all $\alpha \in \mathrm{Seq}$ we have $(A_C)_\alpha = \mathrm{graph}(f_C)$, it is clear that $[A_C X] = f_C(X)$.

(b) Since $A_C \subseteq B_C$, the proposition is clear for $X \not\subseteq C$. Therefore suppose $X \subseteq C$. There are two cases.

Case 1: $X \subsetneq C$. Then there exists a p such that $X \not\supseteq C \cap e_p$. On the other hand we have $(B_C)_{(p)} = \mathrm{graph}(f_C) \cup \mathrm{graph}(g_{p,C})$. By lemma 4.1.7.(ii) we conclude

$$((B_C)_{(p)} \cdot X) = f_C(X) \cup g_{p,C}(X) = \emptyset.$$

Furthermore, if $\mathrm{lth}(\alpha) \neq 1$, then also

$$((B_C)_\alpha \circ X) = f_C(X) = \emptyset.$$

Therefore there exists a path through the tree of sequence numbers, labelled $\emptyset$ at every node. We conclude that

$$[B_C X] = \emptyset.$$

Case 2: $X = C$. Then for *all* $p \in \omega$ we have $X \supseteq C \cap e_p$, hence

$$((B_C)_{(p)} \circ X) = \omega.$$

Therefore $[B_C X] = \omega$.

□

In $\mathbb{P}\omega$ all representable functions are continuous and hence are determined by their applicative behaviour on the finite subsets of ω. Due to the existence of A_C and B_C this is far from true in $\mathbb{H}\omega$. We will state this more formally now.

4.2.6. Definition. Let $\mathfrak{M} = (X, \circ)$ be an applicative structure. $D \subseteq X$ is called *dense (in $\mathfrak{M}$)* if for all $a,b \in X$

$$\forall d \in D(a \circ d = b \circ d) \rightarrow \forall x \in X(a \circ x = b \circ x).$$

□

If we restrict our attention to CPO lambda models $(X, \circ, \leqslant, \perp)$ we observe that a set $D \subseteq X$ is dense in $(X, \circ)$ if and only if all (Scott-) continuous functions from X into X are determined by their values on D. Furthermore, we notice that density as defined above implies topological density with respect to the Scott topology.

4.2.7. Corollary.

(i) The collection of finite subsets of ω is dense in $\mathbb{P}\omega$.

(ii) The only dense subset in $\mathbb{H}\omega$ is the full $P\omega$.

Proof:

(i) By continuity of application and the fact that every set is the directed supremum of its finite subsets.

(ii) Immediate by proposition 4.2.5.

□

4.2.8. Proposition. Let $C,D\subseteq\omega$.

(i) $\forall X\subseteq\omega([A_C X]\subseteq[B_C X])$.

(ii) If $C\subsetneq D$ then $\forall X\subseteq\omega([B_D X]\subseteq[A_C X])$.

Proof:

(i) Since $A_C\subseteq B_C$ this is clear by lemma 4.1.7.(i).

(ii) Suppose $C\subsetneq D$ and let $X\subseteq\omega$.
If $[A_C X] = \omega$ then clearly $[B_D X]\subseteq[A_C X]$.
Therefore assume $[A_C X] = \emptyset$, hence $X\subseteq C$.
Then $X\subsetneq D$ and thus $[B_D X] = \emptyset$.
Also in this case $[B_D X]\subseteq[A_C X]$.
Thus in all cases $[B_D X]\subseteq[A_C X]$. □

By making use of our hypothetical ε under the assumption (MS_ε) we can make these inclusions independent of any $X\subseteq\omega$.

4.2.9. Corollary. Suppose (MS_ε) holds.

(a) $[\varepsilon A_C]\subsetneq[\varepsilon B_C]$, for all $C\subseteq\omega$.

(b) $[\varepsilon B_D]\subseteq[\varepsilon A_C]$, if $C,D\subseteq\omega$, such that $C\subsetneq D$.

Equality holds only if $D=\{d\}$ for some $d\in\omega$ (and hence $C=\emptyset$).

Proof: The inclusions in (a) and (b) follow from proposition 4.2.8. by applying $(MS_\varepsilon 3)$, cf. lemma 4.2.1.

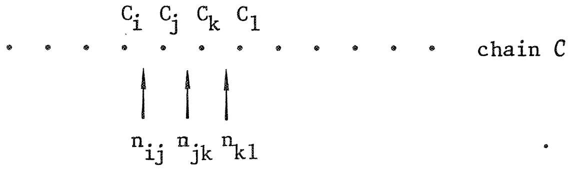
The fact that the inclusion in (a) is strict follows from $(MS_\varepsilon 1)$ and proposition 4.2.5. For, if $[\varepsilon A_C] = [\varepsilon B_C]$, then

$$\emptyset = [A_C C] = [[\varepsilon A_C]C] = [[\varepsilon B_C]C] = [B_C C] = \omega.$$

A contradiction. Hence $[\varepsilon A_C]\neq[\varepsilon B_C]$.

Now assume that D is not a singleton set in order to prove strict inclusion in (b). Let $d\in D\setminus C$. Then $[B_D\{d\}] = \emptyset$ and $[A_C\{d\}] = \omega$. Therefore, again by $(MS_\varepsilon 1)$, $[\varepsilon B_D]\neq[\varepsilon A_C]$. □

Now the crux in the proof of contradicting (MS_ε) lies in the existence and nonexistence of certain chains in $(P\omega,\subseteq)$. It is a quite remarkable fact that, although ω is countable there exists an uncountable chain of subsets of ω. Intuitively this seems to be impossible since whenever C,D are in the chain $\mathcal{C}$, such that say $C \subsetneqq D$, we may choose an element $d \in \omega$ with $d \in D$, $d \notin C$. By doing this for all pairs of sets in the chain we seem to find uncountably many natural numbers. Pictorially the idea is as follows:

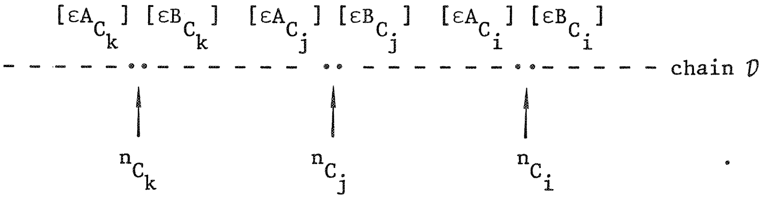


But, as the next lemma shows, it is sometimes impossible to choose uncountably many *different* natural numbers in this way.

4.2.10. <u>Lemma</u>. There exists an uncountable chain $\mathcal{C}$ in $(P\omega,\subseteq)$.

> Proof: Since $\omega \cong \mathbb{Q}$ it is clearly sufficient to give an uncountable chain $\mathcal{C}$ in $(P\mathbb{Q},\subseteq)$. Consider the function $f\colon \mathbb{R} \to P\mathbb{Q}$, given by
>
> $$f(r) = \{q \in \mathbb{Q} \mid q<r\}.$$
>
> Let $\mathcal{C} = \mathrm{Ran}(f)$, then $\mathcal{C}$ is an uncountable chain in $(P\mathbb{Q},\subseteq)$, order-isomorphic to $(\mathbb{R},<)$.
>
> □

Now let $\mathcal{C}$ be this uncountable chain in $(P\omega,\subseteq)$ and assume that (MS_ε) holds. Since $C \mapsto [\varepsilon A_C]$ is a contravariant embedding from $(P\omega,\subseteq)$ into itself, we have another chain $\mathcal{D} = \{[\varepsilon A_C] \mid C \in \mathcal{C}\}$. But this chain has the following peculiar property: for any element $X = [\varepsilon A_C] \in \mathcal{D}$ there exists a companion, namely $[\varepsilon B_C]$, such that this companion is larger than X, but smaller than all $Y \in \mathcal{D}$ such that $X \subsetneqq Y$. This makes it possible to pick natural numbers $n_C \in [\varepsilon B_C]\backslash[\varepsilon A_C]$ for all $C \in \mathcal{C}$, which are all different. Pictorially, we have


$[\varepsilon A_{C_k}]$ $[\varepsilon B_{C_k}]$ $[\varepsilon A_{C_j}]$ $[\varepsilon B_{C_j}]$ $[\varepsilon A_{C_i}]$ $[\varepsilon B_{C_i}]$
chain $\mathcal{D}$
n_{C_k} n_{C_j} n_{C_i}

This idea is worked out in the next theorem.

4.2.11. Theorem. There is no expansion $(P\omega,!,\varepsilon)$ of $\mathbb{H}\omega$ to a lambda model.

Proof: Suppose $(P\omega,!,\varepsilon)$ is a lambda model. Then (MS_ε) is satisfied. By lemma 4.2.10. let $\mathcal{C}$ be an uncountable chain in $(P\omega,\subseteq)$. Define for $C\in\mathcal{C}$

$$n_C = \mu n.n\in[\varepsilon B_C]\setminus[\varepsilon A_C].$$

n_C is welldefined by corollary 4.2.9.(a). Now let us show that the mapping $C\mapsto n_C$ is injective in order to derive a contradiction from the fact that $\mathcal{C}$ is uncountable and ω is countable. Suppose $C,D\in\mathcal{C}$ such that $C\neq D$. Since $\mathcal{C}$ is a chain we have $C\subsetneqq D$ or $D\subsetneqq C$. Assume the former, without loss of generality. Then by corollary 4.2.9.(b) we have $[\varepsilon B_D]\subseteq[\varepsilon A_C]$.
But then

$$n_D\in[\varepsilon B_D]\subseteq[\varepsilon A_C].$$

Hence $n_D\in[\varepsilon A_C]$. Since $n_C\notin[\varepsilon A_C]$ by definition, we conclude $n_C\neq n_D$. This shows the injectivity of the mapping $C\mapsto n_C$ and thereby finishes the proof.

□

REFERENCES

ACZEL, P.

[1980] *Frege structures and the notions of proposition, truth and set.*
Barwise, Keisler and Kunen [1980], pp. 31-59.

ADACHI, T.

[1983] *A categorical characterization of lambda calculus models.*
Research report on Information Sciences C-49, Department of Information Sciences, Tokyo Institute of Technology.

BARENDREGT, H.P.

[1977] *The type free lambda calculus.*
Barwise [1977], pp. 1091-1132.

[1981] *The lambda calculus. Its syntax and semantics.*
Studies in Logic and the Foundations of Mathematics, volume 103. North-Holland, Amsterdam, etc.

BARENDREGT, H.P.; COPPO, M. and DEZANI-CIANCAGLINI, M.

[1983] *A filter lambda model and the completeness of type assignment.*
Journal of Symbolic Logic, volume 48/4, pp. 931-940.

BARENDREGT, H.P. and KOYMANS, K.

[1980] *Comparing some classes of lambda calculus models.*
Seldin and Hindley [1980], pp. 287-301.

BARENDREGT, H.P. and LONGO, G.

[1980] *Equality of lambda terms in the model* T^{ω}.
Seldin and Hindley [1980], pp. 303-337.

BARWISE, J. (editor)

[1977] *Handbook of Mathematical Logic.*
Studies in Logic and the Foundations of Mathematics, volume 90. North-Holland, Amsterdam, etc.

BARWISE, J.; KEISLER, H.J. and KUNEN, K. (editors)

[1980] *The Kleene Symposium* (proceedings).
Studies in Logic and the Foundations of Mathematics, volume 101. North-Holland, Amsterdam, etc.

BERRY, G.

[1980] *On the definition of lambda calculus models.*
Rapport de Recherche 46, Institute National de Recherche en Informatique et en Automatique (INRIA).
Published in Dīaz and Ramos [1981], pp. 218-230.

[1981] *Some syntactic and categorical constructions of lambda calculus models.*
Rapport de Recherche 80, Institute National de Recherche en Informatique et en Automatique (INRIA).

BÖHM, C. (editor)

[1975] *λ-calculus and computer science theory.*
Proceedings of the Rome symposium. Lecture Notes in Computer Science, volume 37. Springer-Verlag, Berlin, etc.

COOPERSTOCK, D.

[1981] *Alternative axiomatizations of models of the lambda calculus.*
Technical Report 151/81, Department of Computer Science, University of Toronto.

CURRY, H.B. and FEYS, R.

[1958] *Combinatory logic, volume I.*
Studies in Logic and the Foundations of Mathematics. North-Holland, Amsterdam, etc.

DĪAZ, J. and RAMOS, I. (editors)

[1981] *International Colloquium on Formalization of Programming Concepts* (proceedings).
Lecture Notes in Computer Science, volume 107. Springer-Verlag, Berlin, etc.

DOLD, A. and ECKMANN, B. (editors)

[1975] *Logic Conference Kiel 1974* (proceedings).
Lecture Notes in Mathematics, volume 499. Springer-Verlag, Berlin, etc.

ENGELER, E.

[1981] *Algebras and combinators.*
Algebra Universalis 13, pp. 389-392.

GIERZ, G.; HOFMANN, K.; KEIMEL, K.; LAWSON, J.D.; MISLOVE, M. and SCOTT, D.S.

[1980] *A compendium of continuous lattices.*
Springer-Verlag, Berlin, etc.

HEIJENOORT, van J. (editor)

[1967] *From Frege to Gödel.*
Harvard University Press, Cambridge.

HINDLEY, R. and LONGO, G.

[1980] *Lambda calculus models and extensionality.*
Zeitschrift für Mathematische Logik und Grundlagen der Mathematik 26, pp. 289-310.

KAROUBI, M.

[1978] *K-theory, an introduction.*
Springer-Verlag, Berlin, etc.

KARPIŃSKI, M. (editor)

[1977] *Fundamentals of computation theory* (proceedings).
Lecture Notes in Computer Science, volume 56. Springer-Verlag, Berlin, etc.

KOYMANS, K.

[1979] *Lambda calculus models, their definition and categorical techniques used in model constructions.*
Master thesis, Department of Mathematics, University of

Utrecht. Manuscript.

[1981] *Models of the lambda calculus.*
Preprint 223, Department of Mathematics, University of Utrecht. Published as Koymans [1982].

[1982] *Models of the lambda calculus.*
Information and Control, volume 52/3, pp. 306-332.

[1983] *Derived λ-calculus models and the construction of D_∞ inside Pω.*
Preprint 283, Department of Mathematics, University of Utrecht.

[1983A] *The hypergraphmodel for combinatory logic is not a lambda model.*
Preprint 288, Department of Mathematics, University of Utrecht.

LAARHOVEN, M.A.M.

[1975] *Getypeerde- en niet getypeerde extensionele λ-calculus-modellen in Pω.*
Master thesis, Onderafdeling der Wiskunde, Technische Hogeschool Eindhoven.

LAWVERE, F.W. (editor)

[1972] *Toposes, Algebraic Geometry and Logic.*
Proceedings of the Dalhousie conference. Lecture Notes in Mathematics, volume 274. Springer-Verlag, Berlin, etc.

LONGO, G. and MOGGI, E.

[1983] *Gödel numberings, principal morphisms, combinatory algebras. A category-theoretic characterization of functional completeness.*
Nota Scientifica S-1983-21 (preliminary draft), Dipartimento di Informatica, Università di Pisa.

MACLANE, S.

[1971] *Categories for the working mathematician.*

Graduate Texts in Mathematics 5. Springer-Verlag, Berlin, etc.

MEYER, A.R.

[1980] *What is a model of the lambda calculus* ?
MIT/LCS/TM-171, Laboratory for Computer Science, Massachusetts Institute of Technology.

[1981] *What is a model of the lambda calculus* ? (expanded version).
MIT/LCS/TM-201, Laboratory for Computer Science, Massachusetts Institute of Technology. Published as Meyer [1982].

[1982] *What is a model of the lambda calculus* ?
Information and Control, volume 52/1, pp. 87-122.

MEYERS, A.

[1974] *Cartesian closed categories and λ-calculi.*
Master thesis, Department of Mathematics, University of Bristol.

MILNE, R. and STRACHEY, C.

[1976] *A theory of programming language semantics*
(part a and part b).
Chapman and Hall, London; Wiley, New York.

NAKAJIMA, R.

[1975] *Infinite normal forms for the λ-calculus.*
Böhm [1975], pp. 62-82.

OBTUŁOWICZ, A.

[1977] *Functorial semantics of the type free λ-βη calculus.*
Karpiñski [1977], pp. 302-307.

[1979] *On the consistent Church algebraic theories, algebraic theories of type λ-βη, and nontrivial models of the type free lambda calculus.*
Institute of Mathematics, Polish Academy of Sciences.
Typescript.

OBTUŁOWICZ, A. and WIWEGER, A.

[1978] *Categorical, functorial and algebraic aspects of the type free lambda calculus.*
Preprint 164, Institute of Mathematics, Polish Academy of Sciences. Published as Obtułowicz and Wiweger [1982].

[1982] *Categorical, functorial and algebraic aspects of the type free lambda calculus.*
Universal Algebra and Applications, Banach Center Publications, volume 9, pp. 399-422. PWN, Polish Scientific Publishers, Warsaw 1982.

PLOTKIN, G.

[1972] *A set-theoretical definition of application.*
Memorandum MIP-R-95, School of Artificial Intelligence, University of Edinburgh.

[1974] *The λ-calculus is ω-incomplete.*
Journal of Symbolic logic, volume 39/2, pp. 313-317.

[1976] *A powerdomain construction.*
SIAM Journal on Computing, volume 5/3, pp. 452-487.

[1978] *T^ω as a universal domain.*
Journal of Computer and System Sciences, volume 17/2, pp. 209-236.

SANCHIS, L.E.

[1979] *Reducibilities in two models for combinatory logic.*
Journal of Symbolic Logic, volume 44/2, pp. 221-234.

SCHÖNFINKEL, M.

[1924] *Über die Bausteine der Mathematischen Logik.*
Mathematische Annalen 92, pp. 305-316. Translated in English as *On the building blocks of Mathematical Logic*, by S. Bauer-Mengelberg, with an introduction by W.V. Quine, in van Heijenoort [1967], pp. 355-366.

SCOTT, D.S.

[1972] *Continuous lattices.*
Lawvere [1972], pp. 97-136.

[1974] *Data types as lattices.*
Dold and Eckmann [1975], pp. 579-651.

[1976] *Data types as lattices.*
Expanded version of Scott [1974].
SIAM Journal on Computing, volume 5/3, pp. 522-587.

[1977] *Logic and programming languages.*
Communications of the ACM, volume 20/9, pp. 634-641.

[1980] *Relating theories of the λ-calculus.*
Seldin and Hindley [1980], pp. 403-450.

[1980A] *Lambda calculus: Some models, some philosophy.*
Barwise, Keisler and Kunen [1980], pp. 223-265.

SELDIN, J.P. and HINDLEY, J.R.

[1980] *To H.B. Curry: Essays on combinatory logic, lambda calculus and formalism.*
Academic Press, London, etc.

YOKOUCHI, H.

[1983] *Cartesian closed structures in models of the lambda calculus.*
Department of Information Sciences, Tokyo Institute of Technology. Typescript.

GLOSSARY OF SYMBOLS

This glossary is divided into the following groups of symbols:

1. *Abbreviations*

2. *Metavariables*
   - *a. General*
   - *b. Syntax*
   - *c. Models and monoids*
   - *d. Categories*

3. *General notations*
   - *a. Sets, functions and sequences*
   - *b. Categories*

4. *Syntax*
   - *a. General*
   - *b. Lambda calculus*
   - *c. Combinatory logic*
   - *d. Axioms and rules*

5. *Models*
   - *a. General*
   - *b. First-order and environmental models*
   - *c. Functional domains and Scott models*
   - *d. Derived models*
   - *e. Graph- en hypergraphmodel*

6. *Categories*
   - *a. Cartesian closed categories*
   - *b. Interpretations in categories*
   - *c. Karoubi construction and retracts*

7. *Monoids*

8. *Complete partial orders, complete and continuous lattices*

9. *Translations and codings.*

1. *Abbreviations*

ca	combinatory algebra	
ccm	cartesian closed monoid	
cλa	combinatory lambda algebra	
cl	complete lattice	
λa,λm	lambda algebra, lambda model	
pλa	proto lambda algebra	
IH	induction hypothesis	
c.c.c.	cartesian closed category	59
cpo	complete partial order	48
λ	lambda calculus	10
po	partial order	48
BA	basic assumption	103
CAT	category of all categories	72
CATES	category of categories with explicit splitting	72
CL	combinatory logic	21
CLATT	category of continuous lattices	7
CPO	category of complete partial orders	49
MS	Meyer-Scott axiom	36
MS$_\varepsilon$	Meyer-Scott axiom for ε	37

2. *Metavariables*

a. General

X,Y,Z,...	sets
a,b,c,...,x,y,z,...	elements of sets
f,g,h,...,φ,ψ,...	functions, maps
n,m,p,...	natural numbers
α,β,γ,...	sequence numbers
A,B,C,...	subsets of ω

b. Syntax

x,y,z,...	variables (for λ and CL)
M,N,L,...	lambda terms

$P,Q,R,\ldots$	combinatory terms (in chapter 3) lambda terms
$\Gamma,\Delta,\Theta,\ldots$	finite sequences of variables

c. Models and monoids

$\mathfrak{M},\mathfrak{N},\ldots$	structures, models
$M,N,\ldots$	monoids
$a,b,c,\ldots$	elements of M as objects in $K(M)$
$m,n,l,\ldots$	elements of M as arrows in $K(M)$
$\rho,\sigma,\ldots$	assignments

d. Categories

$\mathcal{A},\mathcal{B},\mathcal{C},\ldots$	categories
$A,B,C,\ldots$	objects
$f,g,h,\ldots$	arrows
$a,b,c,\ldots$	idempotent arrows
$U,V,\ldots$	reflexive objects

3. *General notations*

a. Sets, functions and sequences

$\lvert\mathfrak{M}\rvert$	domain of the structure $\mathfrak{M}$
ω	set of natural numbers
$\mu n.\varphi(n)$	the smallest $n \in \omega$ such that $\varphi(n)$
PX	power set of X
Y^X	set of functions from X to Y
$X\backslash Y$	$\{x \in X \mid x \notin Y\}$, difference of sets
$X/\!\approx$	X, divided out by the equivalence relation $\approx$
$\mathrm{card}(X)$	cardinality of X
$f \circ g$	composition, f following g
id_X	identity function on X
$\mathrm{Ran}(f)$	range of f
$f \upharpoonright X$	restriction of f to X
$\rightharpoonup$	partial function
$\uparrow$	undefined

$\downarrow$	defined
$\vec{x}$	sequence $x_1,\dots,x_n$
$f(\vec{x}),\overrightarrow{f(x)}$	abbreviation for $f(x_1),\dots,f(x_n)$
$\vec{x}\in X$	abbreviation for $x_1\in X,\dots,x_n\in X$
$\{\vec{x}\}$	abbreviation for $\{x_1,\dots,x_n\}$
$\vec{x}\mapsto f(\vec{x})$	notation for the function f
$\langle\ \rangle$	empty sequence
$\equiv$	identity, identical
$\cong$	isomorphism, isomorphic

b. Categories

$\lvert\mathcal{A}\rvert$	set of objects of $\mathcal{A}$
$\mathcal{A}(A,B)$	set of arrows from A to B
$f\circ g$	composition, f following g
id_A	identity arrow on A
f^{-1}	inverse of f
$\hookrightarrow$	mono(morphism)
$\twoheadrightarrow$	epi(morphism)
$\xrightarrow[=]{\sim}$	iso(morphism)
$\dashv$	adjunction
η	unit of adjunction

4. *Syntax*

a. General

C	set of constants	9
$\underline{X}=\{\underline{a}\mid a\in X\}$	set of constants from X	11
Vars = $\{v_0,v_1,..\}$	variables	9
FV	free variables	9
$M[x:=N]$	substitution	9
$M[\vec{x}:=\vec{N}]$	simultaneous substitution	11
$M(\vec{\underline{a}})$	implicit substitution, $M[\vec{x}:=\vec{\underline{a}}]$	12
$P\restriction d$	restriction of P to the range of d	115
Ass(X)	set of assignments in X	12
$\rho(x/d)$	assignment, changed at one variable	12
I	identity term	35

1	η-expansion of I	35
1_n	n-th η-expansion of I	35
A^nB	iterated function application	92
$AB^{\sim n}$	iterated application to an argument	92
$\vdash$	provability	10,21
T	theory	10,21
$\beta\eta$	theory of βη-equality	135
$=_T$	equivalence modulo provability in T	16,29
$[\ ]_T$	equivalence class modulo $=_T$	16,29
T_1+T_2	union of theories	25
$\mathfrak{B}$	set of Böhm-like trees	136
	theory of Böhm-tree equality	126
$BT(P)$	Böhm-tree	125,135
$NT(P)$	Nakajima-tree	138
$P^{[k]}$	k-th level Böhm-approximation	126
$(A;X)$	η-expansion of Böhm-like tree A	136
$\underset{\sim}{\sqsubset}, {}^{\eta}\underset{\sim}{\sqsubset}, {}^{\eta}\underset{\sim}{\sqsubset}^{\eta}$	Böhm-inclusions	126,127
$\leqslant_\eta$	infinite η-expansion	136
$\infty\eta(A)$	infinite η-normal form	136

b. Lambda calculus

λ	abstraction	9
$\lambda(C)$	lambda calculus with constants	10,11
$\Lambda(C)$	lambda terms with constants	9
$\Lambda^0(C)$	closed lambda terms with constants	11
$\Lambda^0(C,\vec{x})$	lambda terms with variables among $\vec{x}$	11
$\Lambda^0(\mathfrak{M},\vec{x})$	abbreviation for $\Lambda^0(\|\mathfrak{M}\|,\vec{x})$	18
$\lambda,\Lambda,\Lambda^0,\Lambda^0(\vec{x})$	abbreviation for $\lambda(\emptyset),\Lambda(\emptyset),\Lambda^0(\emptyset),\Lambda^0(\emptyset,\vec{x})$	11
$Th(\mathfrak{M})$	theory of $\mathfrak{M}$	11
S_λ,K_λ	basic combinators of λ	20
D_n	n-th canonical derived term	131
M'	interpretation of M in term model	16
$\Lambda\bot$	lambda bottom terms	125
$\mathcal{H}$	theory equating all unsolvables	35
$\mathcal{H}^*$	theory equating all solvably equivalents	127

c. Combinatory logic

d. Axioms and rules

5. *Models*

a. General

$p_{ab}, q_{ab}, ev_{ab}, !_a$	projections, evaluation, unique arrow $a \to T$	95
$\langle f,g \rangle, \Lambda(f)$	pairing, abstraction	95
$\mathfrak{M}(T)$	open term model	16,30
$\mathfrak{M}^0(T)$	closed term model	18,30
$\mathfrak{M}^0 = (X^0, \cdot, [\![\cdot]\!]^0)$	interior	17,30
$\varphi\colon \mathfrak{M} \longrightarrow \mathfrak{N}$	homomorphism	54
$\varphi\colon \mathfrak{M} \hookrightarrow \mathfrak{N}$	embedding	54
$\varphi\colon \mathfrak{M} \xrightarrow{\sim} \mathfrak{N}$	isomorphism	54
$\mathfrak{M} \hookrightarrow \mathfrak{N}$	$\mathfrak{M}$ is embeddable in $\mathfrak{N}$	54
$\mathfrak{M} \cong \mathfrak{N}$	$\mathfrak{M}$ is isomorphic to $\mathfrak{N}$	54
$\mathfrak{M} \twoheadrightarrow \mathfrak{N}$	$\mathfrak{N}$ is homomorphic image of $\mathfrak{M}$	54
$\mathfrak{M}(\mathcal{C})$	λa generated by $\mathcal{C}$	63
$\mathfrak{M}(\mathcal{C},H)$	concrete model given by $\mathcal{C}$,H	98
$\mathfrak{M}(\mathcal{M})$	λa generated by $\mathcal{M}$	95
$\mathcal{K}(\mathfrak{M})$	Karoubi-envelope of $\mathfrak{M}$	95

b. First-order and environmental models

$\mathfrak{M}'$	cλa associated with pλa $\mathfrak{M}$	31
$\mathfrak{N}^+$	pλa associated with cλa $\mathfrak{N}$	31
$\Phi(\mathfrak{M})$	λm associated with ca with ε $\mathfrak{M}$	41
$\Psi(\mathfrak{N})$	ca with ε associated with λm $\mathfrak{N}$	41
s,k	interpretation of S,K	28
$i, 1, 1_n$	interpretation of $I, 1, 1_n$	35
ε	1-like element	37
ε_n^M	Meyer's ε-sequence	37
ε_n^S	Scott's ε-sequence	37
ε_n	either ε_n^M or ε_n^S	38-42
$\bar{s}, \bar{k}$	abbreviations for $\varepsilon_3 s, \varepsilon_2 k$	41

c. Functional domains and Scott models

$[X^n \to X]$	set of representable functions	43
F	function-operator	43
G	graph-operator	43
R	set of functional elements	46
R_n	set of n-th order functional elements	47

d. Derived models

$\mathfrak{M}(d) = (\lvert\mathfrak{M}(d)\rvert, \cdot_d, [\![\cdot]\!]^d)$	derived model, defined by d	111
$\mathfrak{M}(a) = (X(a), F_a, G_a)$	derived CPOλm, defined by a	128
$\mathfrak{M}_n = \mathfrak{M}(d_n)$	n-th canonical derived model	132
$\mathfrak{M}_\infty = \mathfrak{M}(d_\infty)$	limit canonical derived model	134
$\mathbb{P}_n$	n-th canonical derived graphmodel	138
$\mathbb{P}_\infty$	$\mathbb{D}_\infty$ inside $\mathbb{P}\omega$	138
d	derivable element	109
Fix(d)	fixed points of d	109
D_n	n-th canonical derived term	131
d_n	interpretation of D_n	132
d_∞	limit of the d_n's	133
$P \restriction d$	P restricted to the range of d	115
$[\cdot]_\circ$	derived approximation	129
BA	basic assumption	103
(G',F'): $V^V \lhd V$	retract of the derived model	103
δ	arrow representing $r \circ s$	104
d	element representing δ	104
$\mathfrak{M} = \mathfrak{M}(C,U,F,G)$	original model	104
$\mathfrak{N} = \mathfrak{M}(C,V,F',G')$	derived model	104
$\mathcal{F}$: $\mathcal{K}(\mathfrak{N}) \to \mathcal{K}(\mathfrak{M})$	full and faithful embedding	121
(r,s): $V \lhd U$	retract, inducing derived model	102
(ρ,σ): $\lvert\mathfrak{N}\rvert \lhd \lvert\mathfrak{M}\rvert$	retract of domains	108
(G_n,F_n): $U_{n+1} \lhd U_n$	retract of n-th canonical derived model	123-124

e. Hypergraphmodel

$\mathbb{P}\omega$, $\mathbb{H}\omega$	graph-, hypergraphmodel	140
$\cdot$, $!$	graph, hypergraph application	140
(AB),[AB]	graph, hypergraph application	141
Seq	set of sequence numbers	140
$\mathrm{lth}(\alpha)$	length of the sequence α	146
$\alpha \prec \beta$	α is an initial segment of β	140
A_α	α-slice of A	140
$\mathfrak{B}$	bar	140
$\bar{A}$	saturation of A	141

$\bar{\cap}$	saturated intersection	142
$f_C, g_{p,C}$	special continuous functions	144
A_C, B_C	special elements of $\mathbb{H}\omega$	145
D	dense set	146
$\mathcal{C}, \mathcal{D}$	uncountable chains	148

6. *Categories*

a. Cartesian closed categories

$\mathcal{C}$	cartesian closed category	59
$\mathcal{C}\upharpoonright U$	sub-c.c.c., generated by U	95
$A\times B, B^A, T$	product, exponent, terminal	59,60
p_{AB}, p, q_{AB}, q	projections	60
$ev_{AB}, ev, \Lambda_{ABC}, \Lambda$	evaluation and abstraction	60
$!_A$	unique arrow from A to T	59
$\langle f,g\rangle, f\times g$	pairing and product of maps	60
$\pi_x^\Delta, \Pi_\Gamma^\Delta$	generalized projections	62
$[f_1,\ldots,f_n]$	generalized pairing	61
A^n, f^n	iterated product	60,107

b. Interpretations in categories

$(\mathcal{C},U,F,G)$	categorical lambda structure (algebra)	61
U	reflexive object	61
F,G	fun-, graphmorphism	61
$\mathfrak{M}(\mathcal{C})$	λa generated by $\mathcal{C}$	63
$\mathfrak{M}(\mathcal{C},H)$	concrete model induced by $\mathcal{C}$,H	98
$\cdot_A$	application for A-elements	63
$\#\Delta$	length of Δ	62
$\times\Delta$	(#Δ) times iterated product of U	62
$\|\Delta\|$	set of elements of Δ	62
$\Delta\backslash x$	Δ with x left out	62
$\Delta;x$	Δ with x transferred to the back	62
$[\![M]\!]_\Delta$	(intensional) interpretation in a c.c.c.	63
ρ^Δ	assignment as tuple of elements	67

(m,n),$L(m)$	pairing and abstraction in $\mathcal{M}(C)$	80
(a,b),$L(a)$	in general ccm's	82
	in $\mathcal{M}(\mathfrak{M})$	90
$\mathcal{M}(\mathfrak{M})$	lambda monoid induced by λa $\mathfrak{M}$	90
$\mathcal{M}(C)$	ccm induced by c.c.c. C	80
$\mathfrak{M}(\mathcal{M})$	λa induced by lambda monoid $\mathcal{M}$	95
$K(\mathcal{M})$	Karoubi-envelope of monoid $\mathcal{M}$	79
$\widetilde{\mathcal{M}}$	abbreviation for $\mathcal{M}(K(\mathcal{M}))$	87
$\widetilde{L}$	abstraction in $\widetilde{\mathcal{M}}$	87
$\lvert\Delta : M\rvert$	interpretation in a monoid	91
$a \otimes b$	product in a monoid	82
$\widetilde{a}$	restriction of a to paired arguments	82

8. *Complete partial orders, complete and continuous lattices*

cpo	complete partial order	48
$\leqslant$	partial ordering	48
$\bot$	bottom element	48
sup	supremum	48
D	directed set	48
U	Scott open in a cpo	48
	Scott open in a complete lattice	2
ev	evaluation in CPO	49
Λ	abstraction in CPO	49
L	complete lattice	2
D_n	complete lattice in the Scott sequence	2
	cpo in the Scott sequence	53
D_∞	Scott's inverse limit model	2
	Scott's inverse CPO limit model	53
d_n	interpretation of D_n in CPOλm	132
d_∞	limit of the sequence $d_n, n \in \omega$	133
φ_n,ψ_n	retraction pairs in the Scott sequence	2
φ_n,γ_n	retraction pairs in the Scott CPO sequence	53
π_n	projection from D_∞ onto D_n	53
$[X\to Y]$	set of continuous functions from X to Y	2,48

$(x)_n$	approximation in CPOλm	125
$[x]_n$	approximation in derived CPOλm	129

9. *Translations and codings*

λ,CL	standard translations $C\ell \to \Lambda, \Lambda \to C\ell$	23
P_λ, M_{CL}	abbreviation for $\lambda(P)$,CL(M)	23
M^φ, P^φ	image of a term under a homomorphism	54,55
P^d	interpretation of P in the range of d	106
P'	interpretation of P w.r.t. λa inside $\mathcal{K}(\mathfrak{M})$	116
$\langle n,m \rangle$	coding of pairs of natural numbers	50,139
$\langle n,m,p \rangle$	coding of triples of natural numbers	139
e_n	coding of finite subsets of ω	49,139
$(a(0),\ldots,a(k-1))$	coding of finite sequences in ω	139
$\bar{f}(p)$	coding of initial segment of a function	139

INDEX

MATHEMATICAL CENTRE TRACTS

1 T. van der Walt. *Fixed and almost fixed points.* 1963.

2 A.R. Bloemena. *Sampling from a graph.* 1964.

3 G. de Leve. *Generalized Markovian decision processes, part I: model and method.* 1964.

4 G. de Leve. *Generalized Markovian decision processes, part II: probabilistic background.* 1964.

5 G. de Leve, H.C. Tijms, P.J. Weeda. *Generalized Markovian decision processes, applications.* 1970.

6 M.A. Maurice. *Compact ordered spaces.* 1964.

7 W.R. van Zwet. *Convex transformations of random variables.* 1964.

8 J.A. Zonneveld. *Automatic numerical integration.* 1964.

9 P.C. Baayen. *Universal morphisms.* 1964.

10 E.M. de Jager. *Applications of distributions in mathematical physics.* 1964.

11 A.B. Paalman-de Miranda. *Topological semigroups.* 1964.

12 J.A.Th.M. van Berckel, H. Brandt Corstius, R.J. Mokken, A. van Wijngaarden. *Formal properties of newspaper Dutch.* 1965.

13 H.A. Lauwerier. *Asymptotic expansions.* 1966, out of print; replaced by MCT 54.

14 H.A. Lauwerier. *Calculus of variations in mathematical physics.* 1966.

15 R. Doornbos. *Slippage tests.* 1966.

16 J.W. de Bakker. *Formal definition of programming languages with an application to the definition of ALGOL 60.* 1967.

17 R.P. van de Riet. *Formula manipulation in ALGOL 60, part 1.* 1968.

18 R.P. van de Riet. *Formula manipulation in ALGOL 60, part 2.* 1968.

19 J. van der Slot. *Some properties related to compactness.* 1968.

20 P.J. van der Houwen. *Finite difference methods for solving partial differential equations.* 1968.

21 E. Wattel. *The compactness operator in set theory and topology.* 1968.

22 T.J. Dekker. *ALGOL 60 procedures in numerical algebra, part 1.* 1968.

23 T.J. Dekker, W. Hoffmann. *ALGOL 60 procedures in numerical algebra, part 2.* 1968.

24 J.W. de Bakker. *Recursive procedures.* 1971.

25 E.R. Paërl. *Representations of the Lorentz group and projective geometry.* 1969.

26 European Meeting 1968. *Selected statistical papers, part I.* 1968.

27 European Meeting 1968. *Selected statistical papers, part II.* 1968.

28 J. Oosterhoff. *Combination of one-sided statistical tests.* 1969.

29 J. Verhoeff. *Error detecting decimal codes.* 1969.

30 H. Brandt Corstius. *Exercises in computational linguistics.* 1970.

31 W. Molenaar. *Approximations to the Poisson, binomial and hypergeometric distribution functions.* 1970.

32 L. de Haan. *On regular variation and its application to the weak convergence of sample extremes.* 1970.

33 F.W. Steutel. *Preservation of infinite divisibility under mixing and related topics.* 1970.

34 I. Juhász, A. Verbeek, N.S. Kroonenberg. *Cardinal functions in topology.* 1971.

35 M.H. van Emden. *An analysis of complexity.* 1971.

36 J. Grasman. *On the birth of boundary layers.* 1971.

37 J.W. de Bakker, G.A. Blaauw, A.J.W. Duijvestijn, E.W. Dijkstra, P.J. van der Houwen, G.A.M. Kamsteeg-Kemper, F.E.J. Kruseman Aretz, W.L. van der Poel, J.P. Schaap-Kruseman, M.V. Wilkes, G. Zoutendijk. *MC-25 Informatica Symposium.* 1971.

38 W.A. Verloren van Themaat. *Automatic analysis of Dutch compound words.* 1972.

39 H. Bavinck. *Jacobi series and approximation.* 1972.

40 H.C. Tijms. *Analysis of (s,S) inventory models.* 1972.

41 A. Verbeek. *Superextensions of topological spaces.* 1972.

42 W. Vervaat. *Success epochs in Bernoulli trials (with applications in number theory).* 1972.

43 F.H. Ruymgaart. *Asymptotic theory of rank tests for independence.* 1973.

44 H. Bart. *Meromorphic operator valued functions.* 1973.

45 A.A. Balkema. *Monotone transformations and limit laws.* 1973.

46 R.P. van de Riet. *ABC ALGOL, a portable language for formula manipulation systems, part 1: the language.* 1973.

47 R.P. van de Riet. *ABC ALGOL, a portable language for formula manipulation systems, part 2: the compiler.* 1973.

48 F.E.J. Kruseman Aretz, P.J.W. ten Hagen, H.L. Oudshoorn. *An ALGOL 60 compiler in ALGOL 60, text of the MC-compiler for the EL-X8.* 1973.

49 H. Kok. *Connected orderable spaces.* 1974.

50 A. van Wijngaarden, B.J. Mailloux, J.E.L. Peck, C.H.A. Koster, M. Sintzoff, C.H. Lindsey, L.G.L.T. Meertens, R.G. Fisker (eds.). *Revised report on the algorithmic language ALGOL 68.* 1976.

51 A. Hordijk. *Dynamic programming and Markov potential theory.* 1974.

52 P.C. Baayen (ed.). *Topological structures.* 1974.

53 M.J. Faber. *Metrizability in generalized ordered spaces.* 1974.

54 H.A. Lauwerier. *Asymptotic analysis, part 1.* 1974.

55 M. Hall, Jr., J.H. van Lint (eds.). *Combinatorics, part 1: theory of designs, finite geometry and coding theory.* 1974.

56 M. Hall, Jr., J.H. van Lint (eds.). *Combinatorics, part 2: graph theory, foundations, partitions and combinatorial geometry.* 1974.

57 M. Hall, Jr., J.H. van Lint (eds.). *Combinatorics, part 3: combinatorial group theory.* 1974.

58 W. Albers. *Asymptotic expansions and the deficiency concept in statistics.* 1975.

59 J.L. Mijnheer. *Sample path properties of stable processes.* 1975.

60 F. Göbel. *Queueing models involving buffers.* 1975.

63 J.W. de Bakker (ed.). *Foundations of computer science.* 1975.

64 W.J. de Schipper. *Symmetric closed categories.* 1975.

65 J. de Vries. *Topological transformation groups, 1: a categorical approach.* 1975.

66 H.G.J. Pijls. *Logically convex algebras in spectral theory and eigenfunction expansions.* 1976.

68 P.P.N. de Groen. *Singularly perturbed differential operators of second order.* 1976.

69 J.K. Lenstra. *Sequencing by enumerative methods.* 1977.

70 W.P. de Roever, Jr. *Recursive program schemes: semantics and proof theory.* 1976.

71 J.A.E.E. van Nunen. *Contracting Markov decision processes.* 1976.

72 J.K.M. Jansen. *Simple periodic and non-periodic Lamé functions and their applications in the theory of conical waveguides.* 1977.

73 D.M.R. Leivant. *Absoluteness of intuitionistic logic.* 1979.

74 H.J.J. te Riele. *A theoretical and computational study of generalized aliquot sequences.* 1976.

75 A.E. Brouwer. *Treelike spaces and related connected topological spaces.* 1977.

76 M. Rem. *Associons and the closure statement.* 1976.

77 W.C.M. Kallenberg. *Asymptotic optimality of likelihood ratio tests in exponential families.* 1978.

78 E. de Jonge, A.C.M. van Rooij. *Introduction to Riesz spaces.* 1977.

79 M.C.A. van Zuijlen. *Emperical distributions and rank statistics.* 1977.

80 P.W. Hemker. *A numerical study of stiff two-point boundary problems.* 1977.

81 K.R. Apt, J.W. de Bakker (eds.). *Foundations of computer science II, part 1.* 1976.

82 K.R. Apt, J.W. de Bakker (eds.). *Foundations of computer science II, part 2.* 1976.

83 L.S. van Benthem Jutting. *Checking Landau's "Grundlagen" in the AUTOMATH system.* 1979.

84 H.L.L. Busard. *The translation of the elements of Euclid from the Arabic into Latin by Hermann of Carinthia (?), books vii-xii.* 1977.

85 J. van Mill. *Supercompactness and Wallman spaces.* 1977.

86 S.G. van der Meulen, M. Veldhorst. *Torrix I, a programming system for operations on vectors and matrices over arbitrary fields and of variable size.* 1978.

88 A. Schrijver. *Matroids and linking systems.* 1977.

89 J.W. de Roever. *Complex Fourier transformation and analytic functionals with unbounded carriers.* 1978.

90 L.P.J. Groenewegen. *Characterization of optimal strategies in dynamic games.* 1981.

91 J.M. Geysel. *Transcendence in fields of positive characteristic.* 1979.

92 P.J. Weeda. *Finite generalized Markov programming.* 1979.

93 H.C. Tijms, J. Wessels (eds.). *Markov decision theory.* 1977.

94 A. Bijlsma. *Simultaneous approximations in transcendental number theory.* 1978.

95 K.M. van Hee. *Bayesian control of Markov chains.* 1978.

96 P.M.B. Vitányi. *Lindenmayer systems: structure, languages, and growth functions.* 1980.

97 A. Federgruen. *Markovian control problems; functional equations and algorithms.* 1984.

98 R. Geel. *Singular perturbations of hyperbolic type.* 1978.

99 J.K. Lenstra, A.H.G. Rinnooy Kan, P. van Emde Boas (eds.). *Interfaces between computer science and operations research.* 1978.

100 P.C. Baayen, D. van Dulst, J. Oosterhoff (eds.). *Proceedings bicentennial congress of the Wiskundig Genootschap, part 1.* 1979.

101 P.C. Baayen, D. van Dulst, J. Oosterhoff (eds.). *Proceedings bicentennial congress of the Wiskundig Genootschap, part 2.* 1979.

102 D. van Dulst. *Reflexive and superreflexive Banach spaces.* 1978.

103 K. van Harn. *Classifying infinitely divisible distributions by functional equations.* 1978.

104 J.M. van Wouwe. *Go-spaces and generalizations of metrizability.* 1979.

105 R. Helmers. *Edgeworth expansions for linear combinations of order statistics.* 1982.

106 A. Schrijver (ed.). *Packing and covering in combinatorics.* 1979.

107 C. den Heijer. *The numerical solution of nonlinear operator equations by imbedding methods.* 1979.

108 J.W. de Bakker, J. van Leeuwen (eds.). *Foundations of computer science III, part 1.* 1979.

109 J.W. de Bakker, J. van Leeuwen (eds.). *Foundations of computer science III, part 2.* 1979.

110 J.C. van Vliet. *ALGOL 68 transput, part I: historical review and discussion of the implementation model.* 1979.

111 J.C. van Vliet. *ALGOL 68 transput, part II: an implementation model.* 1979.

112 H.C.P. Berbee. *Random walks with stationary increments and renewal theory.* 1979.

113 T.A.B. Snijders. *Asymptotic optimality theory for testing problems with restricted alternatives.* 1979.

114 A.J.E.M. Janssen. *Application of the Wigner distribution to harmonic analysis of generalized stochastic processes.* 1979.

115 P.C. Baayen, J. van Mill (eds.). *Topological structures II, part 1.* 1979.

116 P.C. Baayen, J. van Mill (eds.). *Topological structures II, part 2.* 1979.

117 P.J.M. Kallenberg. *Branching processes with continuous state space.* 1979.

118 P. Groeneboom. *Large deviations and asymptotic efficiencies.* 1980.

119 F.J. Peters. *Sparse matrices and substructures, with a novel implementation of finite element algorithms.* 1980.

120 W.P.M. de Ruyter. *On the asymptotic analysis of large-scale ocean circulation.* 1980.

121 W.H. Haemers. *Eigenvalue techniques in design and graph theory.* 1980.

122 J.C.P. Bus. *Numerical solution of systems of nonlinear equations.* 1980.

123 I. Yuhász. *Cardinal functions in topology - ten years later.* 1980.

124 R.D. Gill. *Censoring and stochastic integrals.* 1980.

125 R. Eising. *2-D systems, an algebraic approach.* 1980.

126 G. van der Hoek. *Reduction methods in nonlinear programming.* 1980.

127 J.W. Klop. *Combinatory reduction systems.* 1980.

128 A.J.J. Talman. *Variable dimension fixed point algorithms and triangulations.* 1980.

129 G. van der Laan. *Simplicial fixed point algorithms.* 1980.

130 P.J.W. ten Hagen, T. Hagen, P. Klint, H. Noot, H.J. Sint, A.H. Veen. *ILP: intermediate language for pictures.* 1980.

131 R.J.R. Back. *Correctness preserving program refinements: proof theory and applications.* 1980.

132 H.M. Mulder. *The interval function of a graph.* 1980.

133 C.A.J. Klaassen. *Statistical performance of location estimators.* 1981.

134 J.C. van Vliet, H. Wupper (eds.). *Proceedings international conference on ALGOL 68.* 1981.

135 J.A.G. Groenendijk, T.M.V. Janssen, M.J.B. Stokhof (eds.). *Formal methods in the study of language, part I.* 1981.

136 J.A.G. Groenendijk, T.M.V. Janssen, M.J.B. Stokhof (eds.). *Formal methods in the study of language, part II.* 1981.

137 J. Telgen. *Redundancy and linear programs.* 1981.

138 H.A. Lauwerier. *Mathematical models of epidemics.* 1981.

139 J. van der Wal. *Stochastic dynamic programming, successive approximations and nearly optimal strategies for Markov decision processes and Markov games.* 1981.

140 J.H. van Geldrop. *A mathematical theory of pure exchange economies without the no-critical-point hypothesis.* 1981.

141 G.E. Welters. *Abel-Jacobi isogenies for certain types of Fano threefolds.* 1981.

142 H.R. Bennett, D.J. Lutzer (eds.). *Topology and order structures, part 1.* 1981.

143 J.M. Schumacher. *Dynamic feedback in finite- and infinite-dimensional linear systems.* 1981.

144 P. Eijgenraam. *The solution of initial value problems using interval arithmetic; formulation and analysis of an algorithm.* 1981.

145 A.J. Brentjes. *Multi-dimensional continued fraction algorithms.* 1981.

146 C.V.M. van der Mee. *Semigroup and factorization methods in transport theory.* 1981.

147 H.H. Tigelaar. *Identification and informative sample size.* 1982.

148 L.C.M. Kallenberg. *Linear programming and finite Markovian control problems.* 1983.

149 C.B. Huijsmans, M.A. Kaashoek, W.A.J. Luxemburg, W.K. Vietsch (eds.). *From A to Z, proceedings of a symposium in honour of A.C. Zaanen.* 1982.

150 M. Veldhorst. *An analysis of sparse matrix storage schemes.* 1982.

151 R.J.M.M. Does. *Higher order asymptotics for simple linear rank statistics.* 1982.

152 G.F. van der Hoeven. *Projections of lawless sequences.* 1982.

153 J.P.C. Blanc. *Application of the theory of boundary value problems in the analysis of a queueing model with paired services.* 1982.

154 H.W. Lenstra, Jr., R. Tijdeman (eds.). *Computational methods in number theory, part I.* 1982.

155 H.W. Lenstra, Jr., R. Tijdeman (eds.). *Computational methods in number theory, part II.* 1982.

156 P.M.G. Apers. *Query processing and data allocation in distributed database systems.* 1983.

157 H.A.W.M. Kneppers. *The covariant classification of two-dimensional smooth commutative formal groups over an algebraically closed field of positive characteristic.* 1983.

158 J.W. de Bakker, J. van Leeuwen (eds.). *Foundations of computer science IV, distributed systems, part 1.* 1983.

159 J.W. de Bakker, J. van Leeuwen (eds.). *Foundations of computer science IV, distributed systems, part 2.* 1983.

160 A. Rezus. *Abstract AUTOMATH.* 1983.

161 G.F. Helminck. *Eisenstein series on the metaplectic group, an algebraic approach.* 1983.

162 J.J. Dik. *Tests for preference.* 1983.

163 H. Schippers. *Multiple grid methods for equations of the second kind with applications in fluid mechanics.* 1983.

164 F.A. van der Duyn Schouten. *Markov decision processes with continuous time parameter.* 1983.

165 P.C.T. van der Hoeven. *On point processes.* 1983.

166 H.B.M. Jonkers. *Abstraction, specification and implementation techniques, with an application to garbage collection.* 1983.

167 W.H.M. Zijm. *Nonnegative matrices in dynamic programming.* 1983.

168 J.H. Evertse. *Upper bounds for the numbers of solutions of diophantine equations.* 1983.

169 H.R. Bennett, D.J. Lutzer (eds.). *Topology and order structures, part 2.* 1983.

CWI TRACTS

1 D.H.J. Epema. *Surfaces with canonical hyperplane sections.* 1984.

2 J.J. Dijkstra. *Fake topological Hilbert spaces and characterizations of dimension in terms of negligibility.* 1984.

3 A.J. van der Schaft. *System theoretic descriptions of physical systems.* 1984.

4 J. Koene. *Minimal cost flow in processing networks, a primal approach.* 1984.

5 B. Hoogenboom. *Intertwining functions on compact Lie groups.* 1984.

6 A.P.W. Böhm. *Dataflow computation.* 1984.

7 A. Blokhuis. *Few-distance sets.* 1984.

8 M.H. van Hoorn. *Algorithms and approximations for queueing systems.* 1984.

9 C.P.J. Koymans. *Models of the lambda calculus.* 1984.