The Lambda Calculus is Algebraic

PETER SELINGER
Department of Mathematics and Statistics
University of Ottawa, Ottawa, Ontario K1N 6N5, Canada
(*e-mail:* `selinger@mathstat.uottawa.ca`)

Abstract

This paper serves as a self-contained, tutorial introduction to combinatory models of the untyped lambda calculus. We focus particularly on the interpretation of free variables. We argue that free variables should not be interpreted as elements in a model, as is usually done, but as indeterminates. We claim that the resulting interpretation is more natural and leads to a closer correspondence between models and theories. In particular, it solves the problem of the notorious ξ-rule, which asserts that equations should be preserved under binders, and which fails to be sound for the usual interpretation.

Introduction

The correspondence between Curry's type-free lambda calculus and Schönfinkel's combinatory algebras is among the oldest known and the most aesthetically pleasing facts about the lambda calculus. However, the combinatory interpretation of the lambda calculus is also known to be somewhat imperfect, because it does not satisfy the ξ-rule: under the interpretation, $M = N$ does not necessarily imply $\lambda x.M = \lambda x.N$ (Barendregt, 1984). Thus, the class of lambda algebras is not sound for lambda theories, and one is forced to consider the non-equational class of lambda models instead. It seems to follow that the lambda calculus does not correspond to an equationally definable class of algebras.

A similar problem arises whenever one tries to model languages that contain binders. Recall that the terms of universal algebra are constructed from variables and n-ary operations from some signature. On the other hand, the terms of most programming languages also contain bound variables. The question arises whether languages with binders are fundamentally more expressive than languages without them, or whether binders are, at least in principle, a dispensable convenience.

There are good reasons for dealing with algebraic languages, rather than languages with binders, under certain circumstances; particularly in connection with equational reasoning. Algebraic languages have simple and well-understood model theories, and the models allow standard constructions such as cartesian products, subalgebras, quotients, and the construction of free algebras.

The above-mentioned problem with the ξ-rule seems to suggest that the lambda calculus is not quite equivalent to an algebraic theory, and thus, that languages with binders are fundamentally more powerful than algebraic languages. In this paper, we take a different

point of view. We suggest another way of looking at the problem, which yields a sense in which the lambda calculus is equivalent to an algebraic theory.

The basic observation is that the failure of the ξ-rule is not a deficiency of the lambda calculus itself, nor of combinatory algebras, but rather it is an artifact of the way free variables are interpreted in a model. Under the usual interpretation, free variables are interpreted as *elements* of a lambda algebra A. Thus, an equation $M = N$ between terms is said to be satisfied if it holds whenever the free variables of M and N are replaced by elements of A. We call this the *local* interpretation. We suggest a different interpretation, called the *absolute* interpretation, under which free variables are interpreted as indeterminates. Let $A[x_1 \ldots x_n]$ be the lambda algebra obtained from A by freely adjoining elements $x_1 \ldots x_n$. Under the absolute interpretation, an equation $M = N$ is said to be satisfied if it holds in $A[x_1 \ldots x_n]$.

The fundamental observation of this paper is that the two interpretations do not coincide, and that the absolute interpretation satisfies all rules of the lambda calculus, including the notorious ξ-rule. It follows that the absolute interpretation is sound and complete with respect to arbitrary lambda theories. Further, we show that the categories of lambda theories and of lambda algebras are equivalent. This, to some extent, justifies the slogan "the lambda calculus is algebraic".

To researchers who specialize in the lambda calculus, the results of this paper are probably well-known, or at least they can be easily derived from "folklore" results. However, to researchers outside this immediate field, these results are not as well-known as they might deserve, and they usually appear only implicitly, if at all, in the published literature. There is still widespread confusion about models of the untyped lambda calculus, particularly about the issues of the ξ-rule, lambda algebras, lambda models, and extensionality. Thus, the main purpose of this paper is to bring these results together in one place, and to present them in a self-contained and accessible way. Where we present background material, Barendregt's monograph usually serves as the standard reference (Barendregt, 1984).

The paper is organized as follows. In Section 1, we summarize the basic theory of combinatory logic and the lambda calculus. In Section 2, we introduce the notion of indeterminates and the absolute interpretation of lambda terms. In Section 3 we consider the $\beta\eta$-case, and in Section 4 we explore analogies with cartesian closed categories.

1 Combinatory models of the lambda calculus

1.1 The lambda calculus and combinatory logic

Definition. Fix a countable set $\mathcal{V}$ of variables. Let C be a set of constants. The sets of *lambda terms* $M, N, \ldots$ and of *combinatory terms* $A, B, \ldots$ are given by the following abstract syntax:

$$\begin{array}{lcl} M, N & ::= & x \mid c \mid MN \mid \lambda x.M \\ A, B & ::= & x \mid c \mid AB \mid \mathbf{K} \mid \mathbf{S} \end{array}$$

Here, x ranges over variables and c ranges over constants. The set of lambda terms thus defined is denoted Λ_C, and the set of combinatory terms is $\mathfrak{C}_C$.

We follow the usual syntactic conventions for lambda terms and combinatory terms

Table 1. *The axioms and rules of the lambda calculus*

$$
\begin{array}{rl}
(\alpha) & M =_\alpha N \Rightarrow M = N \\
(\beta) & (\lambda x.M)N = M[N/x] \\
(\mathit{refl}) & M = M \\
(\mathit{symm}) & M = N \Rightarrow N = M \\
(\mathit{trans}) & M = N,\ N = P \Rightarrow M = P \\
(\mathit{cong}) & M = M',\ N = N' \Rightarrow MN = M'N' \\
(\xi) & M = N \Rightarrow \lambda x.M = \lambda x.N
\end{array}
$$

Table 2. *The axioms and rules of combinatory logic*

$$
\begin{array}{rl}
(k) & \mathbf{K}AB = A \\
(s) & \mathbf{S}ABC = AC(BC) \\
(\mathit{refl}) & A = A \\
(\mathit{symm}) & A = B \Rightarrow B = A \\
(\mathit{trans}) & A = B,\ B = C \Rightarrow A = C \\
(\mathit{cong}) & A = A',\ B = B' \Rightarrow AB = A'B' \\
(\mathit{subst}) & A = A' \Rightarrow A[B/x] = A'[B/x]
\end{array}
$$

(Barendregt, 1984). In particular, application associates to the left and the body of an abstraction extends as far to the right as possible. Thus, we write MNP for $(MN)P$, and $\lambda x.MN$ for $\lambda x.(MN)$.

Free and bound variables are defined as usual, and we write $M =_\alpha N$ if M and N are equal up to renaming of bound variables. Note that a combinatory term never has bound variables; of course it may have free ones. We write $M[N/x]$ for the capture-avoiding substitution of N for x in M. A term with no free variables is called *closed*. We write Λ_C^0 and $\mathfrak{C}_C^0$ for the sets of closed lambda terms and closed combinatory terms, respectively.

Definition. The axioms and rules for deriving equations between lambda terms are shown in Table 1. If $\mathcal{T}$ is a set of equations, we write $\mathcal{T} \vdash M = N$ if $M = N$ is derivable from $\mathcal{T}$ by using these rules. A *lambda theory* is a set of closed equations that is closed under derivability. The unique smallest lambda theory is denoted $\lambda\beta$, and is called the *pure* lambda theory. We write $M =_\beta N$ if $\lambda\beta \vdash M = N$.

Similarly, the axioms and rules of combinatory logic are shown in Table 2. Entailment and theories are defined in the same way as for the lambda calculus. The minimal theory of combinatory logic is denoted CL, and we write $A =_{CL} B$ if $CL \vdash A = B$.

Note that combinatory logic is an algebraic theory in the sense of universal algebra: it is given by an algebraic signature and equations, together with the standard rules of equational reasoning. On the other hand, the lambda calculus is not a priori algebraic in this sense, because of the λ-binder and its corresponding ξ-rule. However, we will show in Section 2 that the lambda calculus is equivalent to an algebraic theory in a suitable sense.

Also note that we did not state the equivalent of the rule (subst) for the lambda calculus. There was no need to do so, since (subst) is derivable from (β), (ξ), (cong), and (refl).

In this section and the next one, we consider only the lambda-β-calculus; the η-rule will not be considered until Section 3.

1.2 Combinatory algebras

Definition. An *applicative structure* $(\mathbf{A}, \cdot)$ is a set $\mathbf{A}$ together with a binary operation "$\cdot$". A *combinatory algebra* $(\mathbf{A}, \cdot, k, s)$ is an applicative structure with distinguished elements k, s satisfying

$$kxy = x \qquad \text{and} \qquad sxyz = xz(yz), \qquad \text{for all } x, y, z \in \mathbf{A}.$$

A *homomorphism of combinatory algebras* is $f\colon \mathbf{A} \to \mathbf{B}$ such that $fk = k$, $fs = s$ and $f(x \cdot y) = fx \cdot fy$, for all $x, y \in \mathbf{A}$.

Example. Let $\mathcal{T}$ be a set of equations between lambda terms. The *open term algebra* $\Lambda_C/\mathcal{T}$ has as its elements the $\mathcal{T}$-equivalence classes of lambda terms, i.e., two terms M, N are considered equal if $\mathcal{T} \vdash M = N$. The operations are defined by $M \cdot N = MN$, $k = \lambda xy.x$, and $s = \lambda xyz.xz(yz)$. The *closed term algebra* $\Lambda_C^0/\mathcal{T}$ is defined analogously.

A more trivial example is given by the open and closed term algebras of combinatory logic: here one can take $k = \mathbf{K}$ and $s = \mathbf{S}$.

Combinatory algebras form an algebraic variety, i.e., they allow the usual algebraic constructions of subalgebras, quotient algebras, free algebras and polynomial algebras $\mathbf{A}[z]$. For instance, $\mathbf{A}[z]$ is standardly constructed as the closed term algebra $\Lambda^0_{\mathbf{A}\cup\{z\}}/\mathcal{T}$, where $\mathcal{T}$ is the set of equations $a \cdot b = c$ which hold in $\mathbf{A}$, for $a, b, c \in \mathbf{A}$. The elements of $\mathbf{A}[z]$ are often called *polynomials* (in one variable) over $\mathbf{A}$.

Definition (Local interpretation of combinatory terms). The terms of combinatory logic can be naturally interpreted in a combinatory algebra. Recall that $\mathfrak{C}_\mathbf{A}$ is the set of combinatory terms with one constant symbol for each element of $\mathbf{A}$. The *local interpretation* $[\![A]\!]_\rho$ of such a term is defined with respect to a valuation of variables $\rho\colon \mathcal{V} \to \mathbf{A}$:

$$[\![x]\!]_\rho = \rho x, \quad [\![c]\!]_\rho = c, \quad [\![\mathbf{K}]\!]_\rho = k, \quad [\![\mathbf{S}]\!]_\rho = s, \quad [\![AB]\!]_\rho = [\![A]\!]_\rho \cdot [\![B]\!]_\rho.$$

We call this the *local* interpretation to distinguish it from the *absolute* interpretation discussed later. For terms $A, B \in \mathfrak{C}_\mathbf{A}$, we say that the equation $A = B$ *holds locally* in $\mathbf{A}$, notation $\mathbf{A} \models_{\mathrm{loc}} A = B$, if for all valuations ρ in $\mathbf{A}$, $[\![A]\!]_\rho = [\![B]\!]_\rho$. If $\mathcal{T}$ is a set of equations, we write $\mathbf{A} \models_{\mathrm{loc}} \mathcal{T}$ if every equation in $\mathcal{T}$ holds locally in $\mathbf{A}$. The following soundness and completeness theorem holds for general reasons of universal algebra:

Proposition 1 (Soundness and completeness for combinatory logic). *Let $\mathcal{T}$ be a set of equations between combinatory terms. For constant-free combinatory terms A and B,*

$\mathcal{T} \vdash A = B$ *iff*
$\mathbf{A} \models_{\mathrm{loc}} A = B$ for all combinatory algebras $\mathbf{A}$ such that $\mathbf{A} \models_{\mathrm{loc}} \mathcal{T}$. $\square$

1.3 The derived lambda abstractor

The significance of the two combinators $\mathbf{K}$ and $\mathbf{S}$ of combinatory logic lies in the fact that they can be used to simulate lambda abstraction. Define $\mathbf{I} = \mathbf{SKK}$. Notice that $\mathbf{I}x =_{CL} x$, for all x. For a combinatory term A and a variable x, define the term $\lambda^* x.A$ inductively:

$$\begin{array}{rcll} \lambda^* x.x & = & \mathbf{I} & \\ \lambda^* x.B & = & \mathbf{K}B & \text{if } x \notin \mathrm{FV}(B), \\ \lambda^* x.BC & = & \mathbf{S}(\lambda^* x.B)(\lambda^* x.C) & \text{otherwise.} \end{array}$$

Note by induction that $(\lambda^* x.A)x =_{CL} A$ holds for any term A. Also, $\mathrm{FV}(\lambda^* x.A) = \mathrm{FV}(A) \setminus \{x\}$. The operation λ^* is called the *derived lambda abstractor* of combinatory logic. It is important to remark here that, in general, the operator λ^* is well-defined only on *terms*, and not on equivalence classes of terms. For this reason, the λ^* operator does not, in general, yield an operator $\lambda^* \colon \mathbf{A}[x] \to \mathbf{A}$, for a combinatory algebra $\mathbf{A}$. We will see in Section 2.2 that we do get such an operator when $\mathbf{A}$ is a lambda algebra.

A consequence of the derived lambda abstractor is combinatory completeness: For every combinatory term A with variables in $x_1, \ldots, x_n$, there exists a closed term f such that $A =_{CL} f x_1 \cdots x_n$. This is achieved by letting $f = \lambda^* x_1 \ldots x_n.A$.

1.4 Interpretation of lambda terms

Using the derived lambda abstractor λ^* of combinatory logic, we can define translations $cl \colon \Lambda_C \to \mathfrak{C}_C$ and $\lambda \colon \mathfrak{C}_C \to \Lambda_C$ from lambda terms to combinatory terms and vice versa:

$$\begin{array}{rcl} x_{cl} & = & x \\ c_{cl} & = & c \\ (MN)_{cl} & = & M_{cl} N_{cl} \\ (\lambda x.M)_{cl} & = & \lambda^* x.M_{cl} \end{array} \qquad \begin{array}{rcl} x_\lambda & = & x \\ c_\lambda & = & c \\ (AB)_\lambda & = & A_\lambda B_\lambda \\ \mathbf{K}_\lambda & = & \lambda xy.x \\ \mathbf{S}_\lambda & = & \lambda xyz.xz(yz) \end{array}$$

Notice that again, these translations are defined on terms, rather than equivalence classes of terms. For example, $(\lambda z.(\lambda x.x)z)_{cl} = \mathbf{S}(\mathbf{KI})\mathbf{I}$ and $(\lambda z.z)_{cl} = \mathbf{I}$ are not equivalent in combinatory logic. Thus, $M =_\beta N$ does not imply $M_{cl} =_{CL} N_{cl}$. The following lemma summarizes the properties that *do* hold. Note that the last part of the lemma follows from the first two parts.

Lemma 2. *For any lambda term M, we have $M_{cl,\lambda} =_\beta M$. For combinatory terms A, B, if $A =_{CL} B$ then $A_\lambda =_\beta B_\lambda$. For lambda terms M, N, if $M_{cl} =_{CL} N_{cl}$, then $M =_\beta N$.* □

We can now interpret lambda terms in any combinatory algebra, by first translating them into combinatory logic via cl:

Definition (Local interpretation of lambda terms). Let $\mathbf{A}$ be a combinatory algebra. For lambda terms $M, N \in \Lambda_{\mathbf{A}}$ and a valuation $\rho \colon \mathcal{V} \to \mathbf{A}$, define

$$\begin{array}{rcl} [\![M]\!]_\rho & = & [\![M_{cl}]\!]_\rho, \\ \mathbf{A} \models_{\mathrm{loc}} M = N & \text{iff} & \mathbf{A} \models_{\mathrm{loc}} M_{cl} = N_{cl}. \end{array}$$

We define $\mathrm{Th}(\mathbf{A})$ to be the set of all closed equations $M = N$ such that $\mathbf{A} \models_{\mathrm{loc}} M = N$. Here, by a closed equation we mean, of course, an equation between closed terms.

This interpretation is not sound for the lambda calculus, since there are derivable equations, such as $\lambda z.(\lambda x.x)z = \lambda z.z$, that do not hold in all combinatory algebras. In particular, $\mathrm{Th}(\mathbf{A})$ need not be a lambda theory!

This leads one to consider the class of lambda algebras, which are precisely those combinatory algebras in which the equations of the $\lambda\beta$-calculus are satisfied.

1.5 Lambda Algebras

Definition. (See (Barendregt, 1984)). A combinatory algebra $\mathbf{A}$ is called a *lambda algebra* if for all combinatory terms $A, B \in \mathfrak{C}_{\mathbf{A}}$,

$$A_\lambda =_\beta B_\lambda \qquad \Rightarrow \qquad \mathbf{A} \models_{\mathrm{loc}} A = B.$$

A *homomorphism of lambda algebras* is a homomorphism of combinatory algebras.

Note that a lambda algebra is a particular kind of combinatory algebra. This is not to be confused with the concept of a "syntactical lambda algebra" of (Hindley & Longo, 1980); see also (Barendregt 1984, p. 101).

Example 3. Let $\mathcal{T}$ be a set of equations between lambda terms. The open term algebra $\Lambda_C/\mathcal{T}$ and the closed term algebra $\Lambda_C^0/\mathcal{T}$ are lambda algebras. In the open terms algebra, $\Lambda_C/\mathcal{T} \models_{\mathrm{loc}} A = B$ iff $\mathcal{T} \vdash A_\lambda = B_\lambda$.

Proposition 4. *The class of lambda algebras can be defined, relative to the class of combinatory algebras, by a set of closed, constant-free equations. In particular, lambda algebras form an algebraic variety.*

Proof
By definition, $\mathbf{A}$ is a lambda algebra if and only if it satisfies all equations $A = B$ where $A, B \in \mathfrak{C}_{\mathbf{A}}$ and $A_\lambda =_\beta B_\lambda$. To prove the claim, notice that we can first remove the constant symbols from A and B by replacing them with fresh variables. We can then eliminate the free variables by applying the derived lambda abstractor. The resulting equations are still valid, and they imply the original ones. □

It is less obvious that the set of equations can be taken to be finite.

Proposition 5 (Curry). *The class of lambda algebras is axiomatized by the equations of combinatory logic and the following five closed equations due to Curry:*

1. $k = s(s(ks)(s(kk)k))(k(skk))$
2. $s = s(s(ks)(s(k(s(ks)))(s(k(s(kk)))s)))(k(k(skk)))$
3. $s(kk) = s(s(ks)(s(kk)(s(ks)k)))(kk)$
4. $s(ks)(s(kk)) = s(kk)(s(s(ks)(s(kk)(skk)))(k(skk)))$
5. $s(k(s(ks)))(s(ks)(s(ks))) = s(s(ks)(s(kk)(s(ks)(s(k(s(ks)))s))))(ks)$

Proof
See (Barendregt, 1984), Thm. 5.2.5. □

The Curry axioms are compact, but not particularly intuitive. We will give another finite axiomatization of lambda algebras in Section 2.4 below.

Remark 6 (Failure of the ξ-rule). By definition, the local interpretation of the lambda calculus in a lambda algebra validates all the equations of the pure lambda calculus. However, this interpretation does not necessarily respect equational *consequences*. In particular, the local interpretation does not in general satisfy the ξ-rule from Table 1: there exist terms M, N and a lambda algebra $\mathbf{A}$ such that $\mathbf{A} \models_{\rm loc} M = N$ but $\mathbf{A} \not\models_{\rm loc} \lambda x.M = \lambda x.N$.

One such example, due to Plotkin, arises when $\mathbf{A}$ is the closed term algebra of the lambda-$\beta\eta$-calculus. By an ingenious construction, Plotkin showed that there exist closed terms M and N such that $MP =_{\beta\eta} NP$ for all *closed* terms P (thus $\mathbf{A} \models_{\rm loc} Mx = Nx$), but $Mx \neq_{\beta\eta} Nx$ (thus $\mathbf{A} \not\models_{\rm loc} \lambda x.Mx = \lambda x.Nx$). The details of Plotkin's construction are out of the scope of this tutorial; the interested reader may look them up in (Plotkin, 1974) or (Barendregt, 1984, Thm. 20.1.1).

The failure of the ξ-rule implies that the local interpretation is not sound with respect to equational consequences. For this reason, we only get a soundness and completeness theorem for the *pure* lambda calculus, i.e., the theory $\lambda\beta$. Note that it is soundness, and not completeness, which is problematic in the general case. We prove a better soundness and completeness theorem in Section 2.3 with respect to a different interpretation of lambda terms.

Theorem 7 (Soundness and completeness for the pure lambda calculus). *For constant-free lambda terms M, N,*

$$\lambda\beta \vdash M = N \quad \text{iff} \quad \mathbf{A} \models_{\rm loc} M = N \text{ for all lambda algebras } \mathbf{A}.$$

Proof
"$\Rightarrow$": By definition of lambda algebras. "$\Leftarrow$": By Example 3 and Lemma 2, the open term algebra $\Lambda/\lambda\beta$ of the lambda beta calculus is a lambda algebra in which $M = N$ iff $M =_\beta N$. □

2 Lambda algebras and indeterminates

2.1 A characterization of $\mathbf{A}[z]$ *for lambda algebras*

Recall that $\mathbf{A}[z]$ is the combinatory algebra obtained from $\mathbf{A}$ by freely adjoining an indeterminate z (in the variety of combinatory algebras). We gave a concrete description of $\mathbf{A}[z]$ in Section 1.2. More abstractly, $\mathbf{A}[z]$ is characterized by the following universal property: $\mathbf{A} \subseteq \mathbf{A}[z]$, and whenever $f\colon \mathbf{A} \to \mathbf{B}$ is a homomorphism of combinatory algebras, and $i \in \mathbf{B}$ is an element, then there exists a unique homomorphism $h : \mathbf{A}[z] \to \mathbf{B}$ which extends f and maps z to i.

If $\mathbf{A}$ is a lambda algebra then so is $\mathbf{A}[z]$. More generally, if $\mathbf{A}$ is a lambda algebra and $f\colon \mathbf{A} \to \mathbf{B}$ is a homomorphism of combinatory algebras, then $\mathbf{B}$ is a lambda algebra. This is because lambda algebras are defined by closed equations (Proposition 4), and closed equations are always preserved by homomorphisms.

For lambda algebras, $\mathbf{A}[z]$ has an interesting explicit description. The following construction is similar to constructions given by Krivine (1993) and, in the case of Curry algebras, by Freyd (1989). Let $\mathbf{A} = (\mathbf{A}, \cdot, k, s)$ be a lambda algebra, and define $\mathbf{B} = (\mathbf{B}, \bullet, K, S)$, where

$$\begin{array}{rcl} \mathbf{B} &=& \{a \in \mathbf{A} \mid a = \mathbf{1}a\}, \quad \text{where } \mathbf{1} = s(ki) \text{ and } i = skk, \\ a \bullet b &=& sab, \\ K &=& kk, \\ S &=& ks. \end{array}$$

Note that ab denotes application in $\mathbf{A}$, and $a \bullet b$ denotes application in $\mathbf{B}$. Also note that $\mathbf{1}ab =_{CL} ab$, and $\mathbf{1}_\lambda =_\beta \lambda xy.xy$. The construction is motivated by considering the elements of $\mathbf{A}[z]$ as given by functions with one additional argument. Application is defined by threading through the extra element, and the constants throw it away:

$$\begin{array}{rcl} (a \bullet b)z &=& (az)(bz), \\ Kz &=& k, \\ Sz &=& s. \end{array}$$

Proposition 8. 1. $\mathbf{B}$ *is a well-defined combinatory algebra.*
2. *The map $\iota\colon \mathbf{A} \to \mathbf{B}$ with $\iota(a) = ka$ is a well-defined homomorphism.*
3. *For every homomorphism $f\colon \mathbf{A} \to \mathbf{C}$ and every $z \in \mathbf{C}$, there is a unique homomorphism $g\colon \mathbf{B} \to \mathbf{C}$ such that $f = g \circ \iota$ and $g(i) = z$. Consequently, $\mathbf{B} \cong \mathbf{A}[z]$.*

For the proof of Proposition 8, we need a lemma:

Lemma 9. *The following hold in any lambda algebra, for elements a, b, c:*

$$\begin{array}{crcl} (a) & \mathbf{1}k &=& k, \\ (b) & \mathbf{1}s &=& s, \\ (c) & \mathbf{1}(ka) &=& ka, \\ (d) & \mathbf{1}(sa) &=& sa, \\ (e) & \mathbf{1}(sab) &=& sab, \\ (f) & s(s(kk)a)b &=& \mathbf{1}a, \\ (g) & s(s(s(ks)a)b)c &=& s(sac)(sbc), \\ (h) & k(ab) &=& s(ka)(kb), \\ (i) & s(ka)i &=& \mathbf{1}a. \end{array}$$

Proof
One easily checks that $(\mathbf{1}k)_\lambda =_\beta k_\lambda$, and similarly for the other equations. □

Proof of Proposition 8
1.: It follows by Lemma 9(a)–(e) that all of k, s, K, S, $a \bullet b$, i and $\mathbf{1}$ are elements of $\mathbf{B}$, for any $a, b \in \mathbf{B}$. In particular, the operations on $\mathbf{B}$ are well-defined. Moreover, Lemma 9(f) and (g) imply that for all $a, b, c \in B$,

$$K \bullet a \bullet b = s(s(kk)a)b = \mathbf{1}a = a,$$
$$S \bullet a \bullet b \bullet c = s(s(s(ks)a)b)c = s(sac)(sbc) = a \bullet c \bullet (b \bullet c).$$

2.: Using Lemma 9(h), we have $\iota(ab) = k(ab) = s(ka)(kb) = \iota(a) \bullet \iota(b)$. Also, clearly $\iota k = K$ and $\iota s = S$.

3.: Define $g(a) = f(a) \cdot z$, and check that this has the desired properties. For uniqueness, take any homomorphism $h\colon \mathbf{B} \to \mathbf{C}$ such that $f = h \circ \iota$ and $h(i) = z$. Then for all $a \in \mathbf{B}$,

$$\begin{array}{rl} h(a) = h(\mathbf{1}a) = h(s(ka)i) & \quad \text{by Lemma 9}(i) \\ = h((ka) \bullet i) = h(ka) \cdot h(i) = h(\iota a) \cdot h(i) = f(a) \cdot z = g(a). & \square \end{array}$$

Corollary 10. *Let $\mathbf{A}$ be a lambda algebra, and let $a, b \in \mathbf{A}$. Then $az = bz$ holds in $\mathbf{A}[z]$ if and only if $\mathbf{1}a = \mathbf{1}b$ holds in $\mathbf{A}$.*

Proof

"⇒": By definition of $\mathbf{A}[z]$, there is a unique map $h\colon \mathbf{A}[z] \to \mathbf{B}$ extending ι and sending z to i. Using Lemma 9(i) twice, we get

$$\mathbf{1}a = s(ka)i = (ka) \bullet i = h(az) = h(bz) = (kb) \bullet i = s(kb)i = \mathbf{1}b.$$

"⇐": If $\mathbf{1}a = \mathbf{1}b$ holds in $\mathbf{A}$, then also in $\mathbf{A}[z]$, thus $az = \mathbf{1}az = \mathbf{1}bz = bz$ in $\mathbf{A}[z]$. □

2.2 Absolute interpretation

Let $M(\bar{x})$ be a lambda term with free variables among $x_1, \ldots, x_n = \bar{x}$. The local interpretation $[\![M]\!]_\rho$, defined in Section 1.2, depends on a valuation $\rho\colon \mathcal{V} \to \mathbf{A}$. Since, in fact, it depends only on the values of ρ at $x_1, \ldots, x_n$, the local interpretation can be viewed as a function $[\![M]\!]^{\bar{x}}_{\rm loc}\colon \mathbf{A}^n \to \mathbf{A}$, sending an n-tuple $\bar{a} \in \mathbf{A}^n$ to $[\![M]\!]_{(\bar{x}:=\bar{a})}$. In these terms, an equation $M = N$ holds locally in $\mathbf{A}$ if M and N define the same function $\mathbf{A}^n \to \mathbf{A}$.

We will now consider a different interpretation of terms, where variables are interpreted as indeterminates, rather than as elements. Specifically, we interpret a term $M(\bar{x})$ as an element in $\mathbf{A}[\bar{x}]$, i.e., as a polynomial, rather than a function. We call this the *absolute* interpretation of M. The absolute interpretation distinguishes more terms than the local one, since, in general, two different polynomials may define the same function.

Definition (Absolute interpretation). The *absolute interpretation* $[\![A]\!]^{\bar{x}}_{\rm abs}$ of a combinatory term $A \in \mathfrak{C}_\mathbf{A}$ with variables among $\bar{x} = x_1, \ldots, x_n$ is an element of $\mathbf{A}[\bar{x}]$, defined as follows:

$$[\![x_i]\!]^{\bar{x}}_{\rm abs} = x_i, \quad [\![c]\!]^{\bar{x}}_{\rm abs} = c, \quad [\![\mathbf{K}]\!]^{\bar{x}}_{\rm abs} = k, \quad [\![\mathbf{S}]\!]^{\bar{x}}_{\rm abs} = s, \quad [\![AB]\!]^{\bar{x}}_{\rm abs} = [\![A]\!]^{\bar{x}}_{\rm abs} \cdot [\![B]\!]^{\bar{x}}_{\rm abs}.$$

Notice that this is the same as the local interpretation $[\![A]\!]_\delta$ under the valuation $\delta\colon \{\bar{x}\} \to \mathbf{A}[\bar{x}]$ that maps each variable x_i to itself. An equation $A = B$ between combinatory terms $A, B \in \mathfrak{C}_\mathbf{A}$ is said to hold *absolutely* in $\mathbf{A}$, written as

$$\mathbf{A} \models_{\rm abs} A = B,$$

if $[\![A]\!]^{\bar{x}}_{\rm abs} = [\![B]\!]^{\bar{x}}_{\rm abs}$, where $FV(A, B) \subseteq \bar{x}$. Notice that, since the canonical homomorphism $\mathbf{A}[\bar{x}] \to \mathbf{A}[\bar{y}]$ is one-to-one for $\bar{x} \subseteq \bar{y}$, this notion is invariant under the addition of dummy variables to $\bar{x}$. For lambda terms $M, N \in \Lambda_\mathbf{A}$, we define

$$\begin{array}{rcl} [\![M]\!]^{\bar{x}}_{\rm abs} & = & [\![M_{cl}]\!]^{\bar{x}}_{\rm abs}, \\ \mathbf{A} \models_{\rm abs} M = N & \text{iff} & \mathbf{A} \models_{\rm abs} M_{cl} = N_{cl}. \end{array}$$

Note that for *closed* terms, the absolute and the local interpretations coincide. In particular, $\mathrm{Th}(\mathbf{A})$, which was defined to be a set of closed equations, is the same for the local and the absolute interpretations. However, the two interpretations yield different equations between open terms.

The terminology "an equation holds absolutely" is motivated by the following lemma. The idea is that a property is "absolute" if it is preserved under homomorphisms.

Lemma 11. *Let $\mathbf{A}$ be a combinatory algebra, and let $A, B \in \mathfrak{C}_\mathbf{A}$ be terms with $\mathrm{FV}(A, B) \subseteq \bar{x}$. The following are equivalent:*

1. $\mathbf{A} \models_{\rm abs} A = B$,
2. $\mathbf{A}[\bar{x}] \models_{\rm loc} A = B$,
3. For all homomorphisms $f\colon \mathbf{A} \to \mathbf{B}$, $\mathbf{B} \models_{\rm loc} A = B$.

Here, $\mathbf{B} \models_{\rm loc} A = B$ is meant in the obvious sense, namely by interpreting constants as their images under f.

Proof
1. ⇒ 3.: Consider $f\colon \mathbf{A} \to \mathbf{B}$ and some valuation $\rho\colon \mathcal{V} \to \mathbf{B}$. Define $g\colon \mathbf{A}[\bar{x}] \to \mathbf{B}$ to be the unique map extending f such that $g(x_i) = \rho(x_i)$ for all i. Then $[\![A]\!]_\rho = g[\![A]\!]_\delta = g[\![B]\!]_\delta = [\![B]\!]_\rho$, which proves $\mathbf{B} \models_{\rm loc} A = B$. 3. ⇒ 2. ⇒ 1.: Trivial. □

Corollary 12. *Absolute validity implies local validity, i.e., $\mathbf{A} \models_{\rm abs} A = B$ implies $\mathbf{A} \models_{\rm loc} A = B$.*

Proof
Lemma 11(3) with f the identity function. □

Lemma 13. *In any lambda algebra, $\mathbf{1}(\lambda^* x.A) = \lambda^* x.A$.*

Proof
By definition of λ^* and Lemma 9(e) and (c). □

The next lemma shows that the ξ-rule, which failed for the local interpretation, is valid for the absolute interpretation.

Lemma 14. *Let $\mathbf{A}$ be a lambda algebra. Let $A, B \in \mathfrak{C}_\mathbf{A}$ be combinatory terms. Then*

$$\mathbf{A} \models_{\rm abs} A = B \qquad \Longleftrightarrow \qquad \mathbf{A} \models_{\rm abs} \lambda^* x.A = \lambda^* x.B$$

Proof
Suppose the variables of A and B are among $x, y_1, \ldots, y_n$. "⇒": Suppose $\mathbf{A}[x, \bar{y}] \models_{\rm loc} A = B$. Then $\mathbf{A}[x, \bar{y}] \models_{\rm loc} (\lambda^* x.A)x = A = B = (\lambda^* x.B)x$, hence by Corollary 10, $\mathbf{A}[\bar{y}] \models_{\rm loc} \mathbf{1}(\lambda^* x.A) = \mathbf{1}(\lambda^* x.B)$. The claim follows by Lemma 13. "⇐": Suppose $\mathbf{A}[x, \bar{y}] \models_{\rm loc} \lambda^* x.A = \lambda^* x.B$. Then $\mathbf{A}[x, \bar{y}] \models_{\rm loc} A = (\lambda^* x.A)x = (\lambda^* x.B)x = B$. □

It follows from this lemma that the derived lambda abstractor $\lambda^* x$ is a well-defined operator $\lambda^* x\colon \mathbf{A}[x] \to \mathbf{A}$ if $\mathbf{A}$ is a lambda algebra. When $\mathbf{A}[x]$ is explicitly constructed as $(B, \bullet, K, S)$ as in Section 2.1, then $\lambda^* x\colon \mathbf{B} \to \mathbf{A}$ turns out to be the map that sends every element a to itself. Using this λ^* operator, the absolute interpretation of a lambda term can be defined directly, i.e., without relying on a translation into combinatory logic:

$$[\![c]\!]^{\bar{x}}_{\rm abs} = c, \quad [\![x_i]\!]^{\bar{x}}_{\rm abs} = x_i, \quad [\![MN]\!]^{\bar{x}}_{\rm abs} = [\![M]\!]^{\bar{x}}_{\rm abs} \cdot [\![N]\!]^{\bar{x}}_{\rm abs}, \quad [\![\lambda x.M]\!]^{\bar{x}}_{\rm abs} = \lambda^* x.[\![M]\!]^{x,\bar{x}}_{\rm abs}.$$

2.3 Soundness and completeness of the absolute interpretation

Proposition 15 (Soundness). *The set of equations that hold absolutely in a lambda algebra $\mathbf{A}$ is closed under the axioms and rules of the lambda calculus. As a consequence, $\mathrm{Th}(\mathbf{A})$ is a lambda theory for any lambda algebra $\mathbf{A}$.*

Proof
Consider each axiom and rule of the lambda calculus from Table 1. (α) and (β) hold in any combinatory algebra, the latter being a simple consequence of the syntactic fact that $(\lambda^* x.A)B =_{CL} A[B/x]$, for combinatory terms A and B. The rules (*refl*), (*symm*), (*trans*) or (*cong*) are trivially satisfied. Finally, the rule (ξ) is satisfied by Lemma 14. □

Theorem 16 (Soundness and completeness for lambda theories). *Let $\mathcal{T}$ be a set of equations between lambda terms. For constant-free lambda terms M and N,*

$$\begin{array}{l}\mathcal{T} \vdash M = N \quad \textit{iff} \\ \mathbf{A} \models_{\text{abs}} M = N \text{ for all lambda algebras } \mathbf{A} \text{ such that } \mathbf{A} \models_{\text{abs}} \mathcal{T}.\end{array}$$

Proof

"$\Rightarrow$": By Proposition 15. "$\Leftarrow$": The open term algebra $\Lambda/\mathcal{T}$ associated with $\mathcal{T}$ is a lambda algebra satisfying $M = N$ iff $\mathcal{T} \vdash M = N$. $\square$

2.4 An alternative axiomatization of lambda algebras

The proofs of Corollary 10, Lemmas 13 and 14, and Proposition 15 do not use the definition of a lambda algebra directly; they only assume that the nine properties of Lemma 9 hold in $\mathbf{A}$ and $\mathbf{A}[\bar{y}]$. In fact, these nine properties already axiomatize the class of lambda algebras.

Lemma 17. *Suppose $\mathbf{A}$ absolutely satisfies the nine properties of Lemma 9. Then for all combinatory terms, $\mathbf{A} \models_{\text{abs}} A_{\lambda,cl} = A$.*

Proof

This is an easy induction; the only interesting cases are the base cases $A = k$ and $A = s$. We first note that for any a, $\mathbf{1}a = s(ka)i = \lambda^*x.ax$, by property ($i$) and the definition of λ^*. For $A = k$, we have $k_{\lambda,cl} = \lambda^*x.\lambda^*y.x = \lambda^*x.kx = \mathbf{1}k = k$ by (a). For $A = s$, we have $s_{\lambda,cl} = \lambda^*xyz.xz(yz) = \lambda^*xyz.sxyz = \lambda^*xy.\mathbf{1}(sxy) = \lambda^*xy.sxy = \lambda^*x.\mathbf{1}(sx) = \lambda^*x.sx = \mathbf{1}s = s$; here, we have used ($b$), ($d$), ($e$), and Lemma 14. $\square$

Theorem 18. *Let $\mathbf{A}$ be a combinatory algebra. Then $\mathbf{A}$ is a lambda algebra if and only if it absolutely satisfies the nine properties of Lemma 9.*

Proof

The left-to-right implication is essentially Lemma 9; note that, since the equations hold in *all* lambda algebras, they therefore hold absolutely. For the converse, if $\mathbf{A}$ absolutely satisfies the nine properties, then the proofs of Corollary 10, Lemmas 13 and 14, and Proposition 15 apply to $\mathbf{A}$. To show that $\mathbf{A}$ is a lambda algebra, assume $A_\lambda =_\beta B_\lambda$. By Proposition 15, $\mathbf{A} \models_{\text{abs}} A_\lambda = B_\lambda$, hence, by definition, $\mathbf{A} \models_{\text{abs}} A_{\lambda,cl} = B_{\lambda,cl}$. By Lemma 17, $\mathbf{A} \models_{\text{abs}} A = B$. $\square$

On their face, the axioms of Lemma 9 appear to be more succinct and more elegant than the Curry axioms of Proposition 5. However, note that our axioms contain free variables, and we require the axioms to hold absolutely. One can eliminate the free variables by applying the derived lambda abstractor to each axiom, but this blows up their size enormously. Thus, we do not beat Curry at his own game, which is to find the most succinct set of *closed* axioms for lambda algebras.

It is worth remarking that the axioms presented in Lemma 9 are not independent; notably, (c) follows from (h), because $\mathbf{1}(ka) = s(ki)(ka) = k(ia) = ka$. Still, we included ($c$) in the list for aesthetic reasons. The author does not know whether the remaining axioms are independent.

2.5 Lambda theories and lambda algebras form equivalent categories

In this section, we define the category of lambda theories, and we show that it is equivalent to the category of lambda algebras.

Definition. The category $\underline{\text{LT}}$ of lambda theories is defined as follows: An object is a pair $\langle C, \mathcal{T}\rangle$, where C is a set of constants and $\mathcal{T}$ a lambda theory in the language Λ^0_C. A *translation* from C to C' is a function $\varphi\colon C \to \Lambda^0_{C'}$. Any such φ extends canonically to a function $\tilde{\varphi}\colon \Lambda^0_C \to \Lambda^0_{C'}$, defined by $\tilde{\varphi}M(c_1,\ldots,c_n) = M(\varphi c_1,\ldots,\varphi c_n)$, where $c_1,\ldots,c_n$ are the constants that appear in M. A morphism from $\langle C,\mathcal{T}\rangle$ to $\langle C',\mathcal{T}'\rangle$ is named by a translation from C to C' such that $\mathcal{T} \vdash M = N$ implies $\mathcal{T}' \vdash \tilde{\varphi}M = \tilde{\varphi}N$ for all $M, N \in \Lambda^0_C$. φ and ψ name the same morphism if $\mathcal{T}' \vdash \tilde{\varphi}M = \tilde{\psi}M$ for all $M \in \Lambda^0_C$. Composition is defined by $\varphi \circ \psi := \tilde{\varphi} \circ \psi$.

Theorem 19. *The category $\underline{\text{LT}}$ of lambda theories is equivalent to the category $\underline{\text{LA}}$ of lambda algebras.*

Proof

We define a pair of functors $F\colon \underline{\text{LT}} \to \underline{\text{LA}}$ and $G\colon \underline{\text{LA}} \to \underline{\text{LT}}$. F maps a lambda theory $\langle C,\mathcal{T}\rangle$ to its closed term algebra $\Lambda^0_C/\mathcal{T}$, which is always a lambda algebra. F maps a morphism $\varphi\colon \langle C,\mathcal{T}\rangle \to \langle C',\mathcal{T}'\rangle$ to the homomorphism $f\colon \Lambda^0_C/\mathcal{T} \to \Lambda^0_{C'}/\mathcal{T}'$ induced by $\tilde{\varphi}\colon \Lambda^0_C \to \Lambda^0_{C'}$. G maps a lambda algebra $\mathbf{A}$ to $\langle \mathbf{A}, \mathrm{Th}(\mathbf{A})\rangle$; note that $\mathrm{Th}(\mathbf{A})$ is a lambda theory by Proposition 15. G maps a homomorphism $f\colon \mathbf{A} \to \mathbf{B}$ to the translation $\varphi\colon \mathbf{A} \to \Lambda^0_{\mathbf{B}}$ with $\varphi a = fa$.

Next, we describe a natural isomorphism $\eta\colon \mathrm{id}_{\underline{\text{LA}}} \to F \circ G$. For every lambda algebra $\mathbf{A}$, define $\eta_{\mathbf{A}} : \mathbf{A} \to F \circ G(\mathbf{A}) = \Lambda^0_{\mathbf{A}}/\mathrm{Th}(\mathbf{A})$ by $\eta_{\mathbf{A}}(a) = a$. This is clearly a homomorphism, and it is natural in $\mathbf{A}$. To see that it is an isomorphism, notice that for every $M \in \Lambda^0_{\mathbf{A}}$ there is a unique $a \in \mathbf{A}$ with $\mathrm{Th}(\mathbf{A}) \vdash M = a$, namely, $a = [\![M]\!]$.

In order to show the desired equivalence of categories, it now suffices to show that F is full and faithful. F is one-to-one on hom-sets by definition of morphisms in $\underline{\text{LT}}$. F is also full: if $f\colon \Lambda^0_C/\mathcal{T} \to \Lambda^0_{C'}/\mathcal{T}'$ is any homomorphism, then f maps a closed lambda term $M(c_1,\ldots,c_n)$ to $M(fc_1,\ldots,fc_n)$, where $c_1,\ldots,c_n$ are the constants that appear in M. This is because M is equivalent to an applicative term made up from $c_1,\ldots,c_n$ and the combinators k and s, which are preserved by f. It follows that $f = F\varphi$, where $\varphi\colon C \to \Lambda^0_{C'}$ is defined by choosing a representative $\varphi(c)$ of $f(c)$, for every $c \in C$. $\square$

2.6 Lambda models

The notion of *lambda model* arises, as in (Barendregt, 1984), if one wishes to prove Proposition 15 with respect to the equations that hold locally. To do this, one needs the "local" equivalent of Lemma 14:

$$\mathbf{A} \models_{\text{loc}} A = B \qquad \Rightarrow \qquad \mathbf{A} \models_{\text{loc}} \lambda^*x.A = \lambda^*x.B.$$

This property is called *weak extensionality*. As we have seen in Remark 6, it does not hold in general. Hence one defines a *lambda model* to be a weakly extensional lambda algebra.

From our point of view, the lambda models are those lambda algebras which are intrinsically local: in a lambda model, an equation holds absolutely if and only if it holds locally.

Or in other words: in a lambda model, every polynomial is determined by its behavior as a function. This property might also be called "well-pointedness", by analogy with category-theoretic language (see Section 4). It characterizes the class of lambda models, as shown in the following proposition. The equivalence of 1. and 2. is due to Meyer and Scott.

Proposition 20. *The following are equivalent for a lambda algebra* $\mathbf{A}$*:*

1. $\mathbf{A}$ is weakly extensional.
2. $\mathbf{A}$ satisfies the so-called Meyer-Scott axiom: for all $a, b \in \mathbf{A}$,
$$\frac{\forall x \in \mathbf{A}. ax = bx}{\mathbf{1}a = \mathbf{1}b}, \quad \text{where } \mathbf{1} = \mathbf{S}(\mathbf{KI}).$$
3. $\mathbf{A}$ is "well-pointed", i.e., every equation that holds locally in $\mathbf{A}$ already holds absolutely.

Proof
1. $\Rightarrow$ 3.: Let $\mathbf{A}$ be weakly extensional and $\mathbf{A} \models_{\mathrm{loc}} A = B$. Let $\bar{x}$ be the list of free variables of A and B. By weak extensionality, $\mathbf{A} \models_{\mathrm{loc}} \lambda^*\bar{x}.A = \lambda^*\bar{x}.B$. This is a closed equation, hence $\mathbf{A} \models_{\mathrm{abs}} \lambda^*\bar{x}.A = \lambda^*\bar{x}.B$, and finally $\mathbf{A} \models_{\mathrm{abs}} A = B$ by Lemma 14.

3. $\Rightarrow$ 2.: Suppose for all $x \in \mathbf{A}$, $ax = bx$. Then $\mathbf{A} \models_{\mathrm{abs}} ax = bx$ by 3., i.e., $ax = bx \in \mathbf{A}[x]$. Hence $\mathbf{1}a = \mathbf{1}b$ by Corollary 10.

2. $\Rightarrow$ 1.: To show weak extensionality, suppose $\mathbf{A} \models_{\mathrm{loc}} A = B$. Then $\mathbf{A} \models_{\mathrm{loc}} (\lambda^*x.A)x = (\lambda^*x.B)x$, hence by 2., $\mathbf{A} \models_{\mathrm{loc}} \mathbf{1}(\lambda^*x.A) = \mathbf{1}(\lambda^*x.B)$, hence by Lemma 13, $\mathbf{A} \models_{\mathrm{loc}} \lambda^*x.A = \lambda^*x.B$. $\square$

Lambda models are less natural than lambda algebras, because they do not form an algebraic variety. Traditionally, they were used for getting by with the local interpretation in proving soundness and completeness theorems, see e.g. (Barendregt, 1984), Thm. 5.2.18. In light of Theorem 16, it is more natural to work with the absolute interpretation. Thus, lambda models are not really needed for interpreting the lambda calculus; they are only interesting as "well-pointed" lambda algebras.

3 The lambda-$\beta\eta$ calculus

Definition. The lambda-$\beta\eta$ calculus is the lambda calculus with the additional axiom

$$(\eta) \quad \lambda x.Mx = M, \quad \text{where } x \notin \mathrm{FV}(M).$$

We write $M =_{\beta\eta} N$ if $M = N$ follows from the axioms in Table 1 and (η). If a lambda theory $\mathcal{T}$ is closed under (η) then it is called a *lambda-$\beta\eta$ theory*.

3.1 Curry algebras

Definition. A *Curry algebra* is a lambda algebra with $\mathbf{1} = \mathbf{I}$ (Lambek, 1980).

Note that Curry algebras form an equational variety.

Proposition 21. *A lambda algebra* $\mathbf{A}$ *is a Curry algebra if and only if* $\mathrm{Th}(\mathbf{A})$ *is a lambda-$\beta\eta$ theory.*

Proof
If $x \notin \mathrm{FV}(M)$, then $\lambda x.Mx =_\beta (\lambda xy.xy)M = \mathbf{1}_\lambda M$. Hence in any Curry algebra, $\lambda x.Mx = \mathbf{1}M = M$. Conversely, if $\mathrm{Th}(\mathbf{A})$ is a lambda-$\beta\eta$ theory, then $\mathbf{A} \models \mathbf{1} = \lambda xy.xy = \lambda x.x = \mathbf{I}$. $\square$

Thus, Curry algebras are to the lambda-$\beta\eta$ calculus what lambda algebras are to the lambda-β calculus.

3.2 Extensional models

An applicative structure is *extensional* if for all $a, b \in \mathbf{A}$,

$$\frac{\forall x \in \mathbf{A}. ax = bx}{a = b}.$$

Extensional combinatory algebras are Curry algebras, and hence models of the lambda-$\beta\eta$-calculus. Extensionality is an intuitive property. However, extensional models do not form an algebraic variety: e.g., the open term algebra of the lambda-$\beta\eta$ calculus is extensional, but the subalgebra of closed terms is not (cf. Remark 6 and (Plotkin, 1974)). In fact, a Curry algebra is extensional iff it is a lambda model, since extensionality is equivalent to the Meyer-Scott axiom in this case.

4 Analogies with cartesian closed categories

Cartesian-closed categories (ccc's) are to the simply-typed lambda calculus what lambda algebras are to the untyped lambda calculus. However, the ξ-rule does not pose any particular problem for interpretations in a ccc. Lambda-abstraction is always a well-defined operation. Why is it that the troublesome ξ-rule is not an issue for the ccc interpretation?

The reason is that the standard interpretation of indeterminates in a ccc corresponds to the absolute, and not the local, interpretation in lambda algebras. A morphism $U^n \to U$ in the category-theoretic interpretation corresponds more closely to an element of $\mathbf{A}[x_1, \ldots, x_n]$ than to a function $\mathbf{A}^n \to \mathbf{A}$ in the algebraic interpretation.

4.1 Ccc models and simply typed lambda calculus

Consider a simply-typed lambda calculus over a fixed set of basic types. In a cartesian closed category $\mathbf{C}$, a simply-typed term $x : \sigma \rhd M : \tau$ is interpreted as a morphism $f \colon A \to B$, where A and B are the interpretations of the types σ and τ, respectively. If $|A| = (1, A)$ denotes the set of morphisms $h : 1 \to A$, then $f \colon A \to B$ gives rise to a function $\hat{f} : |A| \to |B|$ in a natural way. Using our terminology, we will say that an equation $M = N$ holds *locally* if the corresponding morphisms f, g satisfy $\hat{f} = \hat{g}$, i.e., if for all points $h \colon 1 \to A$, $f \circ h = g \circ h$. It holds *absolutely* simply if $f = g$.

In the context of cartesian closed categories, the absolute interpretation is the standard one, whereas the local interpretation is a bit contrived. As in the combinatory case, the local interpretation is not sound; again it is the ξ-rule that is violated. Lambda models are analogous to well-pointed ccc's, i.e., those ccc's in which $\hat{f} = \hat{g}$ implies $f = g$. It is precisely the well-pointed ccc's in which the local and absolute interpretations coincide. However, the class of well-pointed ccc's, just like the class of lambda models, is not algebraic.

The treatment of indeterminates in cartesian closed categories corresponds very closely to our treatment of indeterminates in combinatory algebras. An exponential object B^A can indeed be regarded as a kind of polynomial object $B[x]$, where x is of type A. More precisely, a morphism $D \to B^A$ can be identified with a morphism $D \to B$ in the category $\mathbf{C}[x : A]$ obtained from $\mathbf{C}$ by adding an indeterminate arrow $x : 1 \to A$. For a detailed account of such indeterminate morphisms, see Lambek and Scott (1986).

4.2 Reflexive ccc models

One way of making precise the relationship between lambda algebras and cartesian closed categories is by constructing the former from the latter. This idea is not new; it goes back to Lambek (1980). See also the discussion of C-monoids in (Lambek & Scott, 1986).

Let U be a reflexive object in a cartesian closed category, i.e., U is equipped with morphisms $e\colon U^U \to U$ and $p\colon U \to U^U$ such that $p \circ e = \mathrm{id}_{U^U}$. An untyped lambda term M with free variables $x_1, \ldots, x_n$ is interpreted in the standard way as a morphism $U^n \to U$. Define $\mathbf{A} = (1, U)$ and $a \cdot b = p_* \circ \langle a, b\rangle$, where $p_* : U \times U \to U$ is obtained by uncurrying p.

We say that the object U is *locally well-pointed* if $f \neq g : U \to U$ implies that $f \circ x \neq g \circ x$ for some $x : 1 \to U$.

Proposition 22. 1. *$\mathbf{A} = (\mathbf{A}, \cdot)$ is a lambda algebra.*
2. *$\mathbf{A}$ is a lambda model iff U is locally well-pointed.*
3. *$\mathbf{A}$ is a Curry algebra iff $e \circ p = \mathrm{id}_U$.*
4. $\mathbf{A}[x] \cong (1, U^U) \cong (U, U)$.
5. $\mathbf{A}[x_1, \ldots, x_n] \cong (U^n, U)$.
6. *$\mathbf{A} \models_{\mathrm{loc}} M = N$ iff M, N define the same map $(1, U)^n \to (1, U)$.*
7. *$\mathbf{A} \models_{\mathrm{abs}} M = N$ iff M, N define the same element in (U^n, U).*

Proof
1.: One proves by an easy induction on combinatory terms A that

$$[\![A]\!]_\rho = 1 \xrightarrow{\langle \rho x_1, \ldots, \rho x_n\rangle} U^n \xrightarrow{[\![A_\lambda]\!]_{x_1,\ldots,x_n}} U,$$

where $[\![A]\!]_\rho$ is the interpretation in $\mathbf{A}$, and $[\![A_\lambda]\!]_{x_1,\ldots,x_n}$ is the usual categorical interpretation of A_λ. The result then follows by soundness of the categorical interpretation.

2.–7.: These are straightforward calculations. For 4., use the fact from Section 2.1 that the elements of $\mathbf{A}[x]$ can be identified with those $a \in \mathbf{A}$ such that $\mathbf{1}a = a$. On the other hand, arrows $1 \to U^U$ can be identified with those $a : 1 \to U$ such that $e \circ p \circ a = a$, which is equivalent to $\mathbf{1}a = a$ in $\mathbf{A}$. Moreover, the correspondence respects the natural lambda algebra structure on (U, U). 5. is similar. □

Summary

Algebra is about polynomials and indeterminates as much as it is about signatures and equations. Thus, when looking for algebraic models of a language with variables, it seems

natural to interpret the variables as indeterminates, rather than as elements. In this tutorial, we have examined the issues surrounding the interpretation of free variables in the context of the untyped lambda calculus. We found that the two interpretations do not coincide. Moreover, the interpretation of variables as indeterminates is superior in the sense that it validates the ξ-rule without the need for additional non-algebraic requirements on the model. We conclude that the lambda calculus is algebraic, in the sense that its canonical class of models is the class of lambda algebras.

While we have concentrated on models of the untyped lambda calculus, similar considerations apply to the algebraic modeling of any language with variables and binders. In particular, the same ideas also apply to typed languages. A well-known example is the interpretation of the simply-typed lambda calculus in cartesian-closed categories. In the categorical setting, too, variables are most naturally interpreted as indeterminates. This phenomenon was first described by Lambek and is now considered a standard construction in category theory. However, as we have seen, these ideas are not unique to category theory, or to typed languages, but they apply to algebraic settings in general.

Acknowledgments

I would like to thank the three anonymous referees for their valuable suggestions.

References

Barendregt, H. P. (1984). *The lambda calculus, its syntax and semantics*. 2nd edn. North-Holland.

Freyd, P. J. (1989). Combinators. *Pages 63–68 of: Proceedings of categories in computer science and logic*. Contemporary Mathematics 92. American Mathematical Society.

Hindley, J. R., & Longo, G. (1980). Lambda calculus models and extensionality. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, **26**, 289–310.

Koymans, C. P. J. (1982). Models of the lambda calculus. *Information and control*, **52**, 306–332.

Krivine, J.-L. (1993). *Lambda-calculus, types and models*. Masson.

Lambek, J. (1980). From λ-calculus to cartesian closed categories. *Pages 375–402 of:* (Seldin & Hindley, 1980).

Lambek, J., & Scott, P. J. (1986). *An introduction to higher order categorical logic*. Cambridge Studies in Advanced Mathematics 7. Cambridge University Press, New York.

Meyer, A. R. (1982). What is a model of the lambda calculus? *Information and control*, **52**, 87–122.

Plotkin, G. D. (1974). The λ-calculus is ω-incomplete. *Journal of symbolic logic*, **39**, 313–317.

Seldin, J. P., & Hindley, J. R. (eds). (1980). *To H. B. Curry: Essays on combinatory logic, lambda calculus and formalism*. Academic Press, London, New York.