INFORMATION AND CONTROL 52, 87–122 (1982)

What Is a Model of the Lambda Calculus?*

ALBERT R. MEYER

Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

An elementary, purely algebraic definition of model for the untyped lambda calculus is given. This definition is shown to be equivalent to the natural semantic definition based on environments. These definitions of model are consistent with, and yield a completeness theorem for, the standard axioms for lambda convertibility. A simple construction of models for lambda calculus is reviewed. The algebraic formulation clarifies the relation between combinators and lambda terms.

1. INTRODUCTION

Lambda notation provides a convenient means for writing expressions which denote functions. As an informal example, consider the polynomial expression $x^2 + 7x - 1$. One can construct an expression $\lambda x \,.\, x^2 + 7x - 1$ called a lambda *abstraction* denoting the polynomial function whose values are given by the polynomial expression. Thus, $\lambda x \,.\, x^2 + 7x - 1$ could be read as "the function of x whose value is $x^2 + 7x - 1$," and the defining equation $p(x) = x^2 + 7x - 1$ for the polynomial p could as well be written $p = \lambda x \,.\, x^2 + 7x - 1$. The value of p at the argument 3, for example, is obtainable by *applying* the expression $\lambda x \,.\, x^2 + 7x - 1$ to 3, which entails substituting 3 for x to obtain $3^2 + 7 \cdot 3 - 1$ and evaluating the result to obtain 29. This process of application and evaluation reflects the computational behavior of many modern programming languages—which explains in part the recent interest in the lambda calculus among computer scientists (cf. Landin, 1964, 1965; Plotkin, 1975, 1977; Stoy, 1977; Gordon, 1979).

Some of the power of the lambda calculus is suggested by the way functions of several arguments can be handled. The addition function of two variables, for example, whose value is the sum of the values of the variables, could be denoted $\lambda x \,.\, \lambda y \,.\, x + y$. More accurately, the value of $\lambda x \,.\, \lambda y \,.\, x + y$ is a *functional* which, applied, say, to the argument 2, yields the add-two function of one variable: $\lambda y \,.\, 2 + y$. The add-two function can in turn be

* This work was supported in part by The National Science Foundation, Grants MCS 7719754 and MCS 8010707, and by a grant to the M.I.T. Laboratory for Computer Science by the IBM Corporation.

0019-9958/82/010087-36$02.00/0

applied to the argument 4 to yield the sum 6. So a function of two variables can be regarded as a functional of one variable whose value is a function of one variable, in this case an "add a constant" function. Thus, in studying calculations with lambda notations there is no loss of generality in restricting attention to functions—or more precisely functionals—of one argument, and we shall do so in what follows.

A more intriguing example suggesting the importance and special character of the lambda calculus is the "triple composition" functional T. For any function f of one argument and positive integer n, let $f^{(n)}$ denote the composition $f \circ f \circ \cdots \circ f$ of f with itself n times. The functional T can be defined by the equation $T(f) = f^{(3)}$ or equivalently by $T = \lambda f.\,(\lambda x.\,f(f(f(x))))$. Thus, T applied to the cubic polynomial $\lambda z.\,z^3$ would yield the 27th degree polynomial $\lambda z.\,z^{27}$. By the same reasoning, T applied to T equals the "compose 27 times" functional because $T(T)$ applied to f equals $(T(T))(f) = (T \circ T \circ T)(f) = T(T(T(f))) = T(T(f^{(3)})) = T(f^{(3)} \circ f^{(3)} \circ f^{(3)}) = T(f^{(9)}) = f^{(27)}$.

But although it makes good intuitive sense to define the value of $T(T)$ in this way, there are obvious logical difficulties. Applying a function to itself violates the rules of ordinary set theory which forbid a function from being in its own domain. The violation can quickly lead to contradiction. For example, let P be the "paradoxical" functional such that $P(f)$ is zero if $f(f)$ is not the integer zero, and $P(f)$ is the integer one otherwise. So by definition $P(f) \neq f(f)$ for all f; substituting P for f immediately yields the contradiction $P(P) \neq P(P)$.

The problem we consider is how to give values to such expressions involving functionals which may be applied to themselves. The intuitive sense of examples like $T(T)$ must be preserved while avoiding contradictions from examples like $P(P)$. Such a domain of values would be a *model* for the lambda calculus.

Introduced by (Curry, 1930; Church, 1932/1933), lambda calculus and related systems of combinatory logic have been the subject of research by logicians for more than fifty years. However, the *model theory* of the lambda calculus is a development primarily of the past decade—largely carried out following the lead of Dana Scott. While a host of models and methods for model construction are now available, the clear statement of just what *in general* a model of the lambda calculus may be seems not to be well known. The purpose of this article, which is largely tutorial, is to review briefly why there is an apparent difficulty in defining the notion of model for the lambda calculus, and then to show how this difficulty is overcome.

This question of what a model of the lambda calculus is has bothered me for some time. A similar concern is expressed in (Hindley and Longo, 1980) who comment that "...there seems to be, firstly an assumption that the definition is to obvious to need stating, and secondly a disagreement about

what the definition should be." Reading through the literature describing the various model constructions (Plotkin, 1972, 1978; Wadsworth, 1976; Scott, 1976, 1980(a), (b); Stoy, 1977; Engeler, 1979), I felt as though I kept asking, "What is a group?" and kept being told "Permutations on n letters are a group," or "$\mathbb{Z}_k$ is a group," but was never told that a group is simply an algebraic structure with a binary operation satisfying the well known conditions. It turns out that there *is* a comparably simple definition of model for the lambda calculus. We state the definition now just to verify its simplicity; the reader is not supposed to make detailed sense of it yet. The remainder of this article provides a justification for the claim that the following definition is appropriate.

DEFINITION. A *combinatory algebra* is a structure $\langle D, \cdot \rangle$, where $\cdot$ is a binary operation on D, such that there are elements $K, S \in D$ satisfying

$$(K \cdot d_0) \cdot d_1 = d_0, \tag{1.1}$$

and

$$((S \cdot d_0) \cdot d_1) \cdot d_2 = (d_0 \cdot d_2) \cdot (d_1 \cdot d_2) \tag{1.2}$$

for all $d_0, d_1, d_2 \in D$.

A *combinatory model* of the lambda calculus is a structure $\langle D, \cdot, \varepsilon \rangle$, where $\langle D, \cdot \rangle$ is a combinatory algebra and $\varepsilon \in D$ satisfies

$$(\varepsilon \cdot d_0) \cdot d_1 = d_0 \cdot d_1, \tag{1.3a}$$

$$\text{if} \quad \forall\, d \in D(d_0 \cdot d = d_1 \cdot d) \text{ then } \varepsilon \cdot d_0 = \varepsilon \cdot d_1, \tag{1.3b}$$

for all $d_0, d_1 \in D$, and

$$\varepsilon \cdot \varepsilon = \varepsilon.^{1} \tag{1.3c}$$

Combinatory models serve for what is known as the β-lambda calculus. For the other main variant, known as η-lambda calculus, we simply require that the element ε be a left identity on D, i.e., $\varepsilon \cdot d = d$ for all $d \in D$. Equivalently, we can simplify condition (1.3) to (1.4).

[1] If $\langle D, \cdot \rangle$ is a combinatory algebra and ε satisfies (1.3a, b), then $\varepsilon \cdot \varepsilon$ satisfies (1.3a, b, *and* c); i.e., $\langle D, \cdot, \varepsilon \cdot \varepsilon \rangle$ is a combinatory model, as is easily verified. So in a sense (1.3c) is a redundant, normalizing condition. The reason for requiring it is revealed in the Combinatory Model Theorem (iii) in Section 4.

DEFINITION. An *extensional combinatory model* of the lambda calculus is a combinatory algebra $\langle D, \cdot \rangle$ such that

$$\text{if } \forall\, d \in D(d_0 \cdot d = d_1 \cdot d) \text{ then } d_0 = d_1 \tag{1.4}$$

for all $d_0, d_1 \in D$.

The above definitions are not specially new. The definition of combinatory algebra is due to Curry. Condition (1.4) is known as *extensionality*,[2] and the fact that extensional combinatory algebras serve as models for the η-lambda calculus has been observed often (Barendregt, 1977; Hindley and Longo, 1980; Scott, 1980a). A variant of the definition of combinatory model above is mentioned along the way in (Scott, 1980b). Other slightly more complex but still simple, purely algebraic formulations appear in (Obtulowicz, 1977; Obtulowicz and Wiweger, 1978; Volken, 1978; Aczel, 1980; 1981; Barendregt and Longo, 1980; Barendregt, 1981). Barendregt (1981, Chap. 5, Sect. 4) and Cooperstock (1981) survey many of these.

Nevertheless, it still seems worthwhile to emphasize again here that the general definition of lambda calculus model can be formulated in this elementary way without any of the algebraic baggage—very useful for other purposes—of lattices, continuity, or categories, and also without any of the syntactic baggage of lambda calculus terms. Although the results described are known in one form or another to a number of researchers, I have not seen the story told in quite so elementary a way as attempted below.[3]

To keep this article self-contained, we review in the next section the basic definitions of the syntactic properties of what is known as the *untyped* lambda calculus. It will turn out that most of the standard syntactic results about reductions, normal forms, and Church–Rosser properties will not be needed in our development. The main syntactic notion required is merely that of a *lambda theory*, namely, a system of equations between lambda terms closed under the standard inference rules. (See (Hindley *et al.*, (1972) for a more complete treatment of the syntactic theory and (Barendregt, 1981) as a comprehensive reference.)

Section 3 introduces *environment models*. We develop enough of their properties to explain the view that environment models are the natural, most

[2] Scott (1980a, b) argues persuasively that "extensionality" should more soundly be reserved to refer to the weaker condition (ξ) given in Section 2, but I fear that familiar usage is already too deeply ingrained to adopt Scott's terminology.

[3] (Cooperstock, 1981) is a study similar to this one in which Barendregt's structures, environment models, Aczel structures, Obtulowicz structures, and variations of combinatory models and Scott models (cf. Note 8) are compared. Cooperstock also presents a thoughtful discussion of the sense in which all the structures provide equivalent mechanisms for interpreting lambda terms.

Barendregt (1981) gives a comprehensive treatment of combinatory algebras, lambda models and lambda algebras (cf. Sects. 6, 7).

general formulation of what might be meant by mathematical models for the untyped lambda calculus. In particular, the axioms and rules of inference of lambda calculus are sound when interpreted by environment models; every environment model thus yields an associated lambda theory. The consistency of lambda calculus—a purely syntactic notion commonly proved by syntactic (Church–Rosser) properties—is shown to follow from the existence of nontrivial environment models. The central result is a completeness theorem demonstrating that every lambda theory is the theory associated with some environment model.

The drawback of environment models is that they define purely algebraic conditions by induction on the syntactic structure of lambda terms. In Section 4 we demonstrate the equivalence between the combinatory models defined above and environment models, thereby revealing how to formulate the algebraic conditions needed for models without reference to syntax.

As an easy application of the notion of combinatory model, we will see in Section 5 that the construction of a lambda calculus model in (Engeler, 1979) follows for simpler, more general reasons than was demonstrated there. This completes the main part of the story.

In Sections 6 and 7 we indulge in an algebraic excursion in which several structures akin to combinatory algebras are defined and compared. In Section 6 *lambda models* are introduced; they provide a technically useful variation of combinatory models. In Section 7 the connection between lambda terms and combinatory terms given in Sections 4 and 6 leads to the formulation of *lambda algebras*, which are *not* equivalent either to combinatory algebras or lambda models, but retain the best features of both. In the final section we cite some additional results connecting the model theory and proof theory of lambda calculus. An Appendix contains many of the longer proofs whose presence in the main text would have been a distraction.

2. Syntax and Lambda Theories

We let x, y, z denote variables chosen from some fixed infinite set of variables, d denote a constant, and u, v, w denote *lambda terms* defined inductively as follows. A lambda term is either a variable, a constant, an *application* of the form (uv), or an *abstraction* of the form $(\lambda x\, u)$. For readability the notation $\lambda x\,.\,u$ is usually used for abstractions, and parentheses are omitted in applications with association to the left being understood. Thus, uvw abbreviates $((uv)\,w)$. Occurrences of variables in terms are said to be *bound* or *free* following the usual rules as though λx was a quantifier such as $\exists\, x$. Finally, $\lambda x_1 x_2 \ldots x_n\,.\,u$ abbreviates $\lambda x_1\,.\,\lambda x_2 \ldots .\,\lambda x_n\,.\,u$. For every set C of constants, we let $\Lambda(C)$ denote the

lambda terms whose constants are chosen solely from C; so $\Lambda(\phi)$ denotes the constant-free or *pure* terms.

Let $u[v/x]$ denote the result of substituting the term v for *free* occurrences of x in u subject to the usual provisos about renaming bound variables in u to avoid capture of free variables in v. Two basic axiom schemes (α) and (β), and an optional third axiom scheme (η), reflect the intuition behind abstraction and application.

(α) $(\lambda x\, u) = (\lambda y\, u[y/x])$ for y not free in u (the names of bound variables do not matter),

(β) $((\lambda x\, u)\, v) = u[v/x]$ (application can be computed by substitution),

(η) $(\lambda y\, (uy)) = u$ for y not free in u (every object is a function).

With these axiom schemes we take the usual inference rules for a congruence relation, namely, three rules of inference,

(transitivity and symmetry) $u = v, u = v' \quad \vdash v = v'$,

(congruence) $u = u', v = v' \quad \vdash (uv) = (u'v')$,

(ξ) $u = v \vdash (\lambda x\, u) = (\lambda x\, v)$.

We would also insist on the additional axiom

(reflexivity) $u = u$,

except that it happens to follow already from (β) and (transitivity and symmetry) as the reader may check.

Terms which are provably equal by these from instances of (α), (β) (and (η)), are said to $(\eta$-$)$*convert* to one another. Note that convertibility is an equivalence relation on terms—transitivity and symmetry follow immediately from the corresponding inference rule once reflexivity is proved.

Axioms and inference rules lead directly to the notion of a theory.

DEFINITION. A *lambda theory* $\mathscr{E}$ over a set C of constants is a set of equations between terms in $\Lambda(C)$ containing all instances of (α) and (β), and closed under the rules (transitivity and symmetry), (congruence), and (ξ). The notation $\vdash_{\mathscr{E}} u = v$ means that the equation "$u = v$" is in $\mathscr{E}$. The theory is *extensional* if it also contains all instances of (η).

Clearly, if u converts to v, then $\vdash_{\mathscr{E}} u = v$ for all lambda theories $\mathscr{E}$.

As with convertibility, equality in any lambda theory $\mathscr{E}$ defines an equivalence relation on lambda terms. The $\mathscr{E}$-equivalence class of u is denoted $[\![u]\!]_{\mathscr{E}}$. Namely,

$$[\![u]\!]_{\mathscr{E}} = \{v \mid \vdash_{\mathscr{E}} u = v\}.$$

It is not hard to show that simultaneous substitution preserves equations. That is,

$$\text{if } \vdash_{\mathscr{E}} u_i = v_i \qquad \text{for} \quad i = 0,..., n,$$

$$\text{then } \vdash_{\mathscr{E}} u_0[u_1/x_1,..., u_n/x_n] = v_0[v_1/x_1,..., v_n/x_n],$$

where $u_0[u_1/x_1,..., u_n/x_n]$ denotes the result of *simultaneously* substituting $u_1,..., u_n$ for free occurrences of $x_1,..., x_n$ in u_0. (We require of course that the variables $x_1,..., x_n$ be distinct and, as in the case of (β), that bound variables of u_0 are renamed to avoid capture of free variables in $u_1,..., u_n$.)

This is all we need in the way of syntactic notions about lambda calculus.

3. Values of Terms and Environment Models

The simplest first notion of lambda calculus model might be any set D of *values* together with a mapping from any lambda term u to a value $[[u]] \in D$ such that convertible terms are assigned the same value. Clearly it is a minimal requirement of any notion of model that convertible terms receive the same value in the model. This ought not be the sole requirement of models, however, because it does not guarantee the condition that the value of a term determines its behavior with respect to the values of other terms. Specifically, we expect the inference rules to be *sound*, which leads to the following

Definition. *A (n extensional) value model* of the lambda calculus over a set C of constants consists of a set D whose elements are called *values*, and a mapping $[[\cdot]]$ from $\Lambda(C)$ *onto* D such that

$$[\![(\lambda x\, u)]\!] = [\![(\lambda y\, u[y/x])]\!] \qquad \text{for } y \text{ not free in } u,$$

$$[\![((\lambda x\, u)\, v)]\!] = [\![u[v/x]]\!],$$

$$([\![\lambda y\, (uy)]\!] = [\![u]\!] \text{ for } y \text{ not free in } u),$$

and

$$\text{if } [\![u]\!] = [\![v]\!] \text{ and } [\![u']\!] = [\![v']\!], \text{ then}$$

$$[\![(uu')]\!] = [\![(vv')]\!] \text{ and } [\![(\lambda x\, u)]\!] = [\![(\lambda x\, v)]\!].$$

Clearly, value models are a mere reformulation of lambda theories. Namely, if $\mathscr{E}$ is a lambda theory, then mapping a term to its $\mathscr{E}$-equivalence class yields a value model. Conversely, if $[\![\cdot]\!]$ is the mapping of a value model, then the set of equations "$u = v$" such that $[\![u]\!] = [\![v]\!]$ is the lambda theory which yields the value model.

So although the notion of value model is simple and natural *given* the axioms and rules for lambda calculus, it remains an essentially syntactic notion which hardly serves to justify belief or interest in the axioms. That is, value models fail to capture the intuitive idea of lambda terms as descriptions of functions. (It is like saying that a model of group theory is any assignment of truth values to formulas such that provably equivalent formulas about groups receive the same truth value. The central model-theoretic notion which justifies the rules of proof, namely, the notion of how an algebraic structure satisfies a formula, has been left out.)

What sort of structure allows interpretation of lambda terms? It would be easiest if we could appeal to the standard inductive definition of the value of a term over an ordinary algebraic structure—for example, a structure $\mathscr{C} = \langle D, \cdot \rangle$, where $\cdot$ is a binary operation on the set D. It will be helpful to review how values are determined in this standard case.

We define the set of *$\mathscr{C}$-terms* to be constructed from constants in D, variables, parentheses, and a symbol for the binary operation $\cdot$ on D. Actually for reasons which will appear below, it will be convenient to omit the symbol for $\cdot$ and write (uv) instead of $(u \cdot v)$, so that $\mathscr{C}$-terms become the special case of lambda terms in which lambda abstractions do not occur.

Each $\mathscr{C}$-term has a value in D which is determined as soon as an assignment of values to the free variables in the term is given. That is, the term is thought of as defining a function on D of as many arguments as there are free variables. It turns out to be simpler technically to regard terms as defining functions of *all* the variables, even though the value will actually depend only on the values of the free variables in the term. In the context of lambda calculus, assignments of values to variables are usually referred to as *environments*.

Formally, an environment ρ is any map from the set of all variables into D. The *valuation mapping* $\mathscr{V}_{\mathscr{C}}$ defines for each $\mathscr{C}$-term u, a function $\mathscr{V}_{\mathscr{C}}[u]$ from environments to D. The value of u in the environment ρ is written as $\mathscr{V}[u]\rho$. (We omit the subscript $\mathscr{C}$ whenever it is clear from context.)

The value of a term consisting of a single constant is simply the value of the constant.

$$\mathscr{V}[d]\rho = d \qquad \text{for} \quad d \in D. \tag{3.1}$$

The value of a term consisting of a single variable is the value assigned the variable by an environment.

$$\mathscr{V}[x]\rho = \rho(x). \tag{3.2}$$

Finally, the value in $\mathscr{C}$ of a term of the form (uv) is simply the result of applying the binary operation $\cdot$ to the values of u and v.

$$\mathscr{V}_{\mathscr{C}}[(uv)]\rho = (\mathscr{V}_{\mathscr{C}}[u]\rho) \cdot (\mathscr{V}_{\mathscr{C}}[v]\rho). \tag{3.3}$$

DEFINITION. An equation $u = v$ between $\mathscr{C}$-terms is defined to be *valid* in $\mathscr{C}$, written $\mathscr{C} \vDash u = v$, iff the values of u and v are the same in all environments. Namely,

$$\mathscr{C} \vDash u = v \qquad \text{iff} \quad \mathscr{V}_{\mathscr{E}}[u] = \mathscr{V}_{\mathscr{E}}[v].$$

The difficulty in extending these familiar definitions to lambda terms is that lambda abstractions are meant to denote functions, so their values would not, like ordinary terms, be expected to be *elements* of the structure but rather to be *functions* on it. Of course, once we allow functions as values, a lambda term might be applied to a lambda term whose value was determined to be a function, so we must then admit that the value of a term could also be a *functional* on functions on the structure. Things get even messier if we think of applying a term to itself, for, as we noted in the introduction, this violates the rules of set theory which forbid a function from being in its own domain.

The way out of this potential paradox is quite straightforward (and familiar in recursion theory). Namely, we regard each element over the structure as denoting a function on the structure (much as an integer denotes a partial recursive function via a Godel numbering). So we first require of a model that it consist of a nonempty set D whose elements will be the values of terms, together with a map Φ from D *onto* a set $D \to D$ of certain functions from D to D. We will also want to represent each function in $D \to D$ as an element of D, so we require an inverse map Ψ from $D \to D$ into D (much like a mapping from a recursive function to its least Godel number). That is,

$$D \underset{\Psi}{\overset{\Phi}{\rightleftarrows}} (D \to D),$$

$$f = \Phi(\Psi(f)) \qquad \text{for all} \quad f \in D \to D. \tag{3.4}$$

We shall call the structure $\mathscr{E} = \langle D, \Phi, \Psi \rangle$ a *functional domain.* Note that since Φ maps D *onto* $D \to D$, it follows from Cantor's cardinality theorems that $D \to D$ cannot equal the set of all functions from D to D (except in the trivial case that D has exactly one element). However, we will shortly require that $D \to D$ have enough closure properties to mimic the behavior of the set of all functions from D to D, which is why the suggestive notation $D \to D$ is used.

Now the intended interpretation of an application (uv) is that u denotes a function applied to the argument v. So the value over a functional domain $\mathscr{E}$ of the application is gotten by interpreting the value of u as a function and applying that function to the value of v.

$$\mathscr{V}_{\mathscr{E}}[(uv)]\,\rho = f(\mathscr{V}_{\mathscr{E}}[v]\,\rho), \qquad \text{where } f = \Phi(\mathscr{V}_{\mathscr{E}}[u]\,\rho). \tag{3.5}$$

Finally, we must assign values to lambda abstractions of the form $\lambda x \,.\, u$. The intended interpretation here is that u is an expression which can be evaluated for any given value of x, and $\lambda x \,.\, u$ denotes the function f whose value $f(d)$ is obtained from the evaluation of u when x is assigned the value $d \in D$. However, since we want values of terms to be elements rather than functions, we define the value of the abstraction to be the element $\Psi(f) \in D$ which represents the function f. To describe the assignment of d to x, let $\rho\{d/x\}$ denote an environment which agrees with ρ at all variables other than x and which assigns x the value d.

$$(3.6) \quad \mathscr{V}[\lambda x \,.\, u]\rho = \Psi(f), \qquad \text{where } f: D \to D \text{ is the function such that } f(d) = \mathscr{V}[u](\rho\{d/x\}).$$

The only possible catch in clause (3.6) is that the function f may not be in the set $D \to D$, in which case $\Psi(f)$ is undefined. We take the denial of this possibility as our fundamental definition of model.

DEFINITION. An *environment model*[4] of the lambda calculus is any functional domain such that if values are assigned to lambda terms according to (3.1), (3.2), (3.5), and (3.6) above, the functions $f = \lambda d \in D.\mathscr{V}[u](\rho\{d/x\})$ arising in (3.6) are all in $D \to D$.

An equation $u = v$ between lambda terms is defined to be *valid* in an environment model $\mathscr{E}$, written $\mathscr{E} \models u = v$, iff the values of u and v are the same in all environments. Namely,

$$\mathscr{E} \models u = v \qquad \text{iff} \quad \mathscr{V}_{\mathscr{E}}[u] = \mathscr{V}_{\mathscr{E}}[v].$$

[4] The technical setup here is very close to that of (Wadsworth, 1976), except that I have dropped any requirement of a lattice structure on D as well as the requirement that the maps Φ and Ψ be (continuous) isomorphisms. In fact (Obtulowicz, 1977; Obtulowicz and Wiweger, 1978) give essentially this definition which they credit as implicit in (Wadsworth, 1976). Precisely the definition of environment model is also given in (Cooperstock, 1981), where it is credited as jointly proposed by Cooperstock and C. Rackoff based on the preceding earlier references.

Barendregt (1977) defines valuations over a more general class of structures resembling functional domains using essentially the same rules (3.1–2), (3.5–6), but valuations over these more general structures suffer the flaw that the (ξ) rule is not sound. More recently Berry (1980) has offered a definition which is a combination of Barendregt's notion and value models.

The pathologies of Barendregt's structures are lucidly analyzed by Hindley and Longo (1978) who essentially identify the structures as combinatory algebras and lambda algebras (considered in Section 7). Hindley and Longo also arrive at a definition equivalent to environment models (which they call λ-structures) by adding to Barendregt's formulation the requirement that the (ξ) rule be sound. They note by the way that their formulation was obtained independently of Barendregt's, and I note that my definitions were formulated independently of all the papers subsequent to (Wadsworth, 1976).

To illustrate this definition, consider the term $\lambda x_1 x_2 . x_1$ which intuitively denotes the first projection function of two variables. Let p_1 be the value of $\lambda x_1 x_2 . x_1$ in some environment ρ. For $d_1, d_2 \in D$, let $d_1 d_2$ abbreviate $(\Phi(d_1))(d_2)$, and let $d_1 d_2 \cdots d_n$ be read as associated to the left, i.e., as $(\ldots((d_1 d_2) d_3)\ldots d_n)$. Then we expect p_1 to have the property that $p_1 d_1 d_2 = d_1$ for all $d_1, d_2 \in D$.

To verify this, observe that $p_1 d_1 = (\mathscr{V}[\lambda x_2 . x_1])(\rho\{d_1/x_1\})$ by (3.6). Then $p_1 d_1 d_2 = \mathscr{V}[x_1]((\rho\{d_1/x_1\})\{d_2/x_2\})$ by (3.6) again, and the righthand side of this equation equals d_1 by (3.2).

A more general technical justification of the reasonableness of the definition of environment model comes from the fact that the axioms and rules of lambda calculus already follow from the definition. To show this, we begin by observing, following (Wadsworth, 1976), that our use of environments propertly reflects the properties of substitution in formulas. First, the value of a term depends only on the values of its free variables.

FREE VARIABLE LEMMA. $\mathscr{V}[u]\rho = \mathscr{V}[u](\rho\{d/y\})$ *for* y *not free in* u.

More generally, we have the

SUBSTITUTION LEMMA. $\mathscr{V}[u[v/x]]\rho = \mathscr{V}[u](\rho\{d/x\})$ *for* $d = \mathscr{V}[v]\rho$.

Both lemmas are proved by routine induction on the structure of lambda terms. We omit the proof (cf. Stoy, 1977). The Substitution Lemma leads to the fundamental

SOUNDNESS THEOREM. *The equations valid in an environment model form a lambda theory. In particular, if* u *converts to* v, *then* $u = v$ *is valid in all environment models.*

A simple application of the notion of environment model is a model-theoretic proof of the syntactic consistency of lambda calculus, viz., nonconvertibility between some pair of terms.

LEMMA. *The equation* $\lambda x_1 \ldots x_n . x_i = \lambda x_1 \ldots x_n . x_j$, *where* $1 \leqslant i < j \leqslant n$, *is not valid in any environment model with more than one element.*

Proof. For $1 \leqslant k \leqslant n$ and any environment ρ, let $p_k = \mathscr{V}[\lambda x_1 \ldots x_n . x_k]\rho$. From the definition of $\mathscr{V}$, it follows as in the example above that $p_k d_1 \ldots d_n = d_k$ for all $d_1, \ldots, d_n \in D$. If D has more than one element, there exist $d_1, \ldots, d_n \in D$ such that $d_i \neq d_j$, so that $p_i d_1 \ldots d_n = d_i \neq d_j = p_j d_1 \ldots d_n$ and therefore $p_i \neq p_j$. ∎

An immediate consequence of this lemma is that every *nontrivial* model, i.e., model with more than one element, is infinite. Assuming, as we show in Section 5, that nontrivial environment models exist, the preceding lemma and the Soundness Theorem immediately imply the

CONSISTENCY THEOREM. *For* $1 \leqslant i < j \leqslant n$ *the term* $\lambda x_1 \ldots x_n . x_i$ *does not convert to* $\lambda x_1 \ldots x_n . x_j$.

The preceding definitions can be easily modified to deal with η-conversion. The intuitive content of the η-rule is that an element may be identified with the function it specifies, so that distinct elements must specify distinct functions. This amounts simply to the

DEFINITION. An *extensional environment model* is an environment model for which the map $\Phi : D \to (D \to D)$ is one-to-one.

We then can easily show the

η-SOUNDNESS THEOREM. *The equations valid in an extensional environment model form an extensional lambda theory. In particular, if* u η-*converts to* v, *then* $u = v$ *is valid in all extensional environment models.*

The Consistency Theorem similarly extends to extensional environment models and η-conversion.

A converse to the Soundness Theorem provides the most important technical support for the argument that environment models correctly capture the intuitive meaning of lambda calculus as embodied in the convertibility rules. The axioms and rules of lambda calculus provide a *complete* logical system for proving equations about environment models.

COMPLETENESS THEOREM. *Every lambda theory consists of precisely the equations valid in some environment model. That is, for every lambda theory* $\mathscr{E}$, *there is an environment model* $\mathscr{E}$ *such that*

$$\vdash_{\mathscr{E}} u = v \qquad \textit{iff} \quad \mathscr{E} \models u = v.$$

In particular, $u = v$ *is valid in all environment models iff* u *converts to* v.

The required environment model is constructed from classes of provably equivalent terms in much the way that polynomial domains are formally constructed from a ring of coefficients.

DEFINITION. Let $\mathscr{E}$ be a lambda theory over a set C of constants. The *term model associated* with $\mathscr{E}$ is the environment model $\langle D, \Phi, \Psi \rangle$, where

$$D = \{[\![u]\!]_{\mathscr{E}} \mid u \in \Lambda(C)\},$$

$$(\Phi([\![u]\!]_{\mathscr{E}}))([\![v]\!]_{\mathscr{E}}) = [\![(uv)]\!]_{\mathscr{E}},$$

and

$$\Psi(\Phi([\![u]\!]_{\mathscr{E}})) = [\![\lambda x . ux]\!]_{\mathscr{E}} \qquad \text{for} \quad x \text{ not free in } u.$$

The proof that the term model is well defined and is indeed an environment model appears in the Appendix.

4. Combinatory Models

The notion of environment model may best reflect the intuitively correct way to assign values to terms of the lambda calculus, but it is mathematically a bit awkward. The condition that all the functions f arising in (3.6) be in $D \to D$ obviously defines some kind of closure condition on this set of functions, but the formulation of the condition is so entangled with the syntax of lambda terms that it is hard to visualize what models look like, and it can be awkward to verify that particular functional domains are indeed models.

Is there some way to define the closure conditions implicit in (3.6) without reference to the syntactic machinery of lambda terms? Not surprisingly, a solution lies in considering combinators, which were originally devised to short-circuit the syntactic complexities of variables in terms. *Combinators* are simply variable free terms over combinatory algebras.

The key property which motivates the rather odd definition of combinatory algebra given in the introduction is revealed by considering the more natural notion of combinatory completeness defined below. A structure is combinatorially complete if every function definable by a term is equal to the function defined by left multiplication by some constant. More precisely,

Definition. Let $\cdot$ be a binary operation on a set D. The structure $\mathscr{C} = \langle D, \cdot \rangle$ is *combinatorially complete* iff for every $\mathscr{C}$-term u and every sequence $x_1, \ldots, x_n$ of variables containing all the variables in u, there is a constant $d \in D$ such that $\mathscr{C} \models u = dx_1 \cdots x_n$.

Definition. Let I abbreviate the combinator SKK. For every $\mathscr{C}$-term u and variable x define a new $\mathscr{C}$-term $\langle x \rangle u$ as follows:

$$\langle x \rangle u = Ku \qquad \text{if } x \text{ is does not occur in } u,$$

$$\langle x \rangle x = I,$$

$$\langle x \rangle (uv) = S(\langle x \rangle u)(\langle x \rangle v) \qquad \text{if } x \text{ does occur in } u \text{ or } v.$$

Combinatory Completeness Lemma (Curry). *A structure $\mathscr{C} = \langle D, \cdot \rangle$ is combinatorially complete iff it is a combinatory algebra.*

Proof. ($\leftarrow$) Let $\mathscr{C}$ be a combinatory algebra and K, $S \in D$ satisfy (1.1) and (1.2). It follows directly from the definitions that x does not occur in

$\langle x\rangle u$, and that $\mathscr{C} \vDash u = (\langle x\rangle u)x$. Let d be the value in $\mathscr{C}$ of the combinator $\langle x_1\rangle(\ldots(\langle x_n\rangle u))$. Then $\mathscr{C} \vDash u = dx_1 \ldots x_n$ as required.

($\rightarrow$) This is left to the reader. ∎

We now show that the simple definition of combinatory models given at the outset provides the desired algebraic characterization of environment models.

DEFINITION. Let $\mathscr{C} = \langle D, \cdot, \varepsilon\rangle$ be a combinatory model. Let Φ map elements of D into the functions from D to D defined by left multiplication. That is, let $(\Phi(d_0))(d) = d_0 \cdot d$. The *functional domain associated with* $\mathscr{C}$ is $\langle D, \Phi, \Psi\rangle$, where Ψ is given by the rule $\Psi(\Phi(d)) = \varepsilon \cdot d$.

Note that (1.3b) implies that Ψ is well defined, and (1.3a) implies that Ψ is a right inverse of Φ, so that $\langle D, \Phi, \Psi\rangle$ is indeed a functional domain.

DEFINITION. Let $\mathscr{E} = \langle D, \Phi, \Psi\rangle$ be an environment model. Define a binary operation $\cdot$ on D by the rule $d_0 \cdot d_1 = (\Phi(d_0))(d_1)$, and let $\varepsilon = \mathscr{V}_{\mathscr{E}}[\lambda xy\,.\,xy]\,\rho$. The *algebra associated with* $\mathscr{E}$ is $\langle D, \cdot, \varepsilon\rangle$.

Note that by the Free Variable Lemma, the value of ε does not depend on the environment ρ.

COMBINATORY MODEL THEOREM. (i) *The functional domain* $\mathscr{E}$ *associated with a combinatory model* $\mathscr{C}$ *is an environment model which assigns the same values to* $\mathscr{C}$*-terms. That is,*

$$\mathscr{V}_{\mathscr{E}}[u] = \mathscr{V}_{\mathscr{E}}[u]$$

for all $\mathscr{C}$*-terms* u.

(ii) *The algebra associated with an environment model is a combinatory model.*

(iii) *The associations between combinatory models and environment models defined above are inverses of each other. That is, if* $\mathscr{E}$ *is the environment model associated with a combinatory model* $\mathscr{C}$, *then* $\mathscr{C}$ *is the combinatory model associated with* $\mathscr{E}$, *and vice versa.*

The final argument in support of our claim that environment models and combinatory models are alternative formulations of the same class of objects is that the additional expressive power apparently provided by lambda terms in the context of environment models is still achievable in the context of combinatory models. In fact, there is a simple, effective translation from any lambda term u into an equivalent $\mathscr{C}$-term $u^{(\mathscr{C})}$.

To state the relation between the values of lambda terms in an environment model and $\mathscr{C}$-terms in the associated combinatory model, we

establish for lambda terms a result corresponding to combinatorial completeness. Let K_λ abbreviate $\lambda xy\,.\,x$, S_λ abbreviate $\lambda xyz\,.\,xz(yz)$, and I_λ abbreviate $((S_\lambda K_\lambda) K_\lambda)$. Define *combinatory lambda terms* inductively to be lambda terms which are either constants, variables, K_λ, S_λ, or applications of combinatory lambda terms. In other words, combinatory lambda terms are precisely those lambda terms in which the only abstractions occurring are in K_λ and S_λ.

COMBINATORY LAMBDA TERM LEMMA. *For every $u \in \Lambda(C)$ there is a combinatory lambda term $u^{(\lambda)} \in \Lambda(C)$ such that u converts to $u^{(\lambda)}$.*

Proof. For any *combinatory* lambda term $u \in \Lambda(C)$ and variable x, define $\langle\langle x\rangle\rangle\, u \in \Lambda(C)$ as follows:

$$\langle\langle x\rangle\rangle\, u = (K_\lambda u) \qquad \text{if } x \text{ is not free in } u,$$

$$\langle\langle x\rangle\rangle = I_\lambda,$$

$$\langle\langle x\rangle\rangle(uv) = ((S_\lambda \langle\langle x\rangle\rangle\, u)\langle\langle x\rangle\rangle\, v) \qquad \text{if } x \text{ is free in } u \text{ or } v.$$

It follows by simple calculation from the definitions that $\langle\langle x\rangle\rangle\, u$ is a combinatory lambda term which converts to $\lambda x\,.\,u$.

Let $u^{(\lambda)}$ to be the combinatory lambda term obtained from an *arbitrary* lambda term u by replacing all occurrences of λx by $\langle\langle x\rangle\rangle$. Inductively, we may define $u^{(\lambda)}$ as follows:

$$d^{(\lambda)} = d, \qquad x^{(\lambda)} = x, \qquad (uv)^{(\lambda)} = (u^{(\lambda)} v^{(\lambda)}),$$

$$(\lambda x\,.\,u)^{(\lambda)} = \langle\langle x\rangle\rangle(u^{(\lambda)}). \quad \blacksquare$$

DEFINITION. Let $\mathscr{C} = \langle D, \cdot, \varepsilon\rangle$ be a combinatory model. For any $u \in \Lambda(D)$, define $u^{(\mathscr{C})}$ be the $\mathscr{C}$-term obtained by replacing all occurrences of K_λ and S_λ in $u^{(\lambda)}$ by the constant values in D of K_λ and S_λ in the environment model associated with $\mathscr{C}$.

Note that by the Free Variable Lemma, the values of the closed terms K_λ and S_λ are determined by the environment model alone and not by any particular choice of environment. So $u^{(\mathscr{C})}$ is well defined.

COMBINATORY MODEL THEOREM. (iv) Let $\mathscr{E}$ be an environment model and $\mathscr{C}$ its associated combinatory model. Then for all lambda terms u,

$$\mathscr{V}_{\mathscr{E}}[u] = \mathscr{V}_{\mathscr{E}}[u^{(\mathscr{C})}].$$

Proof. $\mathscr{V}_{\mathscr{E}}[u] = \mathscr{V}_{\mathscr{E}}[u^{(\lambda)}]$ by Soundness since u converts to $u^{(\lambda)}$. The proof now follows immediately by induction on the number of occurrences of free variables, constants, and terms K_λ and S_λ in $u^{(\lambda)}$. $\blacksquare$

The Combinatory Model Theorem demonstrates that combinatory models and environment models are merely notational variants of the same class of mathematical structures. (This correspondence between the two kinds of models could be reformulated more abstractly in terms of effective functorial bijections between categories, but I do not think such a formulation helps in this case.)

The significance of the these results is that we can now straightforwardly interpret arbitrary lambda terms and equations between them as though they were standard terms and equations over an ordinary algebraic structure defined by first order axioms.

5. An Elementary Model Construction

We now present a simple construction of a class of combinatory models using only elementary properties of sets.

Let A be any nonempty set, and let B be the least set containing A and all ordered pairs consisting of a finite subset $\beta \subseteq B$ and an element $b \in B$. Such an ordered pair is denoted $(\beta \to b)$. Assume that elements of A are distinguishable from ordered pairs.

Let $D_A = 2^B = \{d \mid d \subseteq B\}$, and define the binary operation $\cdot$ on D_A by the rule

$$d_1 \cdot d_2 = \{b \in B \mid (\beta \to b) \in d_1 \text{ for some } \beta \subseteq d_2\}. \tag{5.1}$$

Model Existence Theorem (Plotkin, 1972).[5] *The structure* $\langle D_A, \cdot, \varepsilon \rangle$ *is a combinatory model where*

$$\varepsilon = \{(\alpha \to (\beta \to b)) \mid \alpha, \beta \text{ finite subsets of } B \text{ and } b \in \alpha \cdot \beta\}.^6 \tag{5.2}$$

[5] The construction is taken directly from (Engeler, 1979). It is a notational variant of one of several models first described in (Plotkin, 1972). These constructions are nearly the same as the better known $P\omega$ construction (Scott, 1976).

Indeed, Longo (personal communication, 1981) has shown that D_A and $P\omega$ have the same pure lambda theory and each is isomorphically embeddable in the other. However, they define different set theoretic *inclusions* among the values of the pure closed terms, and their binary operations behave differently; i.e., they are not even isomorphic as combinatory algebras.

[6] The choice of ε is not unique. For example, let

$$\varepsilon^+ = \varepsilon \cup A \cup \{(\beta \to a) \mid \beta \subseteq B \text{ and } a \in A\}.$$

Then $\langle D_A, \cdot, \varepsilon \rangle$ and $\langle D_A, \cdot, \varepsilon^+ \rangle$ are distinct expansions of the combinatory algebra $\langle D_A, \cdot \rangle$ to combinatory models. Longo (personal communication, 1981) has even shown that they have distinct pure lambda theories; in fact ε^+ yields an interesting model which, in contrast to the D_A model with ε or the $P\omega$ model, does not give all unsolvable terms the same value (cf. Section 8).

Proof. Choose

$$K = \{(\alpha \to (\beta \to b)) | b \in \alpha\},$$

and

$$S = \{(\alpha \to (\beta \to (\gamma \to b))) | b \in \alpha \cdot \gamma \cdot (\beta \cdot \gamma)\}. \tag{5.4}$$

The proof that ε, K, and S given by (5.2–4) satisfy (1.1), (1.2), (1.3a,c) is a direct consequence of the definitions and is omitted. To verify (1.3b), note that $A \cap \varepsilon d_0 = \varnothing$ and that $b \in d_0 \cdot \beta$ iff $(\beta \to b) \in \varepsilon d_0$ by definition. Hence, if $\varepsilon d_0 - \varepsilon d_1 \neq \varnothing$, then $(\beta \to b) \in \varepsilon d_0 - \varepsilon d_1$ for some $(\beta \to b)$, so $b \in d_0 \cdot \beta - d_1 \cdot \beta$. ∎

The model Existence Theorem validates the assumption made in proving the Consistency Theorem in Section 3 that nontrivial models exist. Indeed, D_A has uncountably many elements. There are also countably infinite models; e.g., the recursively enumerable elements of D_A form a countable submodel. (The Lambda Algebra Theorem in Section 7 provides another mechanism for constructing a nontrivial countable model from any nontrivial combinatory model.)

Now let $\times$ be any binary operation on the set A. Let functions f_n and f from A to D_A be defined as follows:

$$\begin{aligned} f_0(a) &= \{a\}, \\ f_{n+1}(a) &= f_n(a) \cup \{(\{a'\} \to b) | a' \in A, b \in f_n(a \times a')\}, \\ f(a) &= \bigcup_{n \geqslant 0} f_n(a). \end{aligned} \tag{5.5}$$

EMBEDDING THEOREM (Engeler, 1979). The function f given in (5.5) isomorphically embeds the structure $\langle A, \times \rangle$ into $\langle D_A, \cdot \rangle$.

Proof. Note that $f(a) \cap A = \{a\}$, so f is injective. The verification that f is a homomorphism is a routine calculation which we omit (cf. Engeler, 1979). ∎

To illustrate the significance of the Embedding Theorem, we can now make sense of the examples involving integers, polynomials, and triple composition given in the Introduction. For example, to obtain the piecewise integer polynomials as part of a combinatory model, let A be the least set containing the integers $\mathbb{Z}$ and distinct new elements add, add_n, $mult$, $mult_n$, $cond$, $cond_a$, $cond_{a,a'}$ for $n \in \mathbb{Z}$, $a, a' \in A$. Define a binary operation $\times$ on A by the rules

$$add \times n = add_n, \qquad add_n \times m = n + m,$$
$$mult \times n = mult_n, \qquad mult_n \times m = nm,$$
$$cond \times a = cond_a, \qquad cond_a \times a' = cond_{a,a'},$$
$$cond_{a,a'} \times a'' = a' \text{ if } a \in \mathbb{N}, \text{ otherwise } a'',$$

for $n, m \in \mathbb{Z}$, and $a, a', a'' \in A$; the operation $\times$ may be defined arbitrarily on arguments not specified above.

Now embedding $\langle A, \times \rangle$ into D_A yields (an isomorphic copy of) $\mathbb{Z}$ along with addition, multiplication, and conditionals. The triple composition functional T of the Introduction also appears in D_A, since $T = \lambda fx.f(f(fx))$ is defined by a pure lambda expression and so can be interpreted in D_A without even appealing to the Embedding Theorem. The reader is invited to consider why the difficulties surrounding the paradoxical functional P no longer threaten.

For construction of extensional models, see Scott (1980a, Sect. 5) who sketches an elementary construction of an embedding theorem into extensional models based on a modification of D_A. Scott (1981) provides another construction of extensional embeddings based on a universal embedding property for the $P\omega$ model. See Wadsworth (1977) for a detailed treatment of Scott's original construction of extensional models from continuous lattices.

6. Another Algebraic Axiomatization: Lambda Models

The definition of combinatory model connects nicely with the definition of environment model, but suffers the small technical disadvantage that the elements K, S are not identified uniquely. For example, $\{(\{b\} \to (\varnothing \to b)) | b \in B\}$ is another element in D_A distinct from the K of (5.3) which satisfies (1.1). In fact, there are uncountably many elements in D_A which satisfy (1.1).

In order to maintain a full correspondence between the algebraic properties of a combinatory model and the values of lambda terms, it is necessary to make the appropriate choice of K and S, namely, as the values of K_λ and S_λ in the associated environment model. As an exercise we shall now show how to describe these values in a purely algebraic way. Then we define Lambda Models essentially as combinatory models in which K, S are properly chosen. The Lambda Model Theorem given below summarizes how to determine K, S from ε and *vice versa*. Readers who are not amused by this kind of exercise in algebra should skip directly to the Lambda Model Theorem and the discussion following it.

Let $\mathscr{C}$ be a combinatory model, $\mathscr{E}$ its associated environment model, and begin by choosing *any* K, S satisfying (1.1) and (1.2). Let

$$B = S(KS)K \tag{6.1}$$

be the "composition" combinator. It is easy to verify that $Bxyz = x(yz)$ is valid in any combinatory algebra. Let

$$\varepsilon_2 = (B \cdot \varepsilon) \cdot (B \cdot \varepsilon). \tag{6.2}$$

We now have

$$\mathscr{V}_{\mathscr{E}}[K_\lambda] = \varepsilon_2 \cdot K, \tag{6.3}$$

because

$$\begin{aligned}\mathscr{V}_{\mathscr{E}}[\lambda xy\,.\,x]\rho &= \Psi(\lambda d\,.\,\mathscr{V}_{\mathscr{E}}[\lambda y\,.\,x]\,\rho\{d/x\}) = \Psi(\lambda d\,.\,\Psi(\lambda e\,.\,d)) \\ &= \Psi(\lambda d\,.\,\Psi(\Phi(K \cdot d))) = \Psi(\lambda d\,.\,\varepsilon \cdot (K \cdot d)) \\ &= \Psi(\Phi((B \cdot \varepsilon) \cdot K)) = \varepsilon \cdot ((B \cdot \varepsilon) \cdot K) = ((B \cdot \varepsilon) \cdot (B \cdot \varepsilon)) \cdot K.\end{aligned}$$

Note that as predicted by the Free Variable Lemma, the value of the closed term K_λ is determined by the environment model $\mathscr{E}$ alone and *not* by any particular choice of ρ, K, or S.
Letting

$$\varepsilon_3 = (B \cdot \varepsilon) \cdot (B \cdot \varepsilon_2), \tag{6.4}$$

a similar calculation shows that

$$\mathscr{V}_{\mathscr{E}}[S_\lambda] = \varepsilon_3 \cdot S. \tag{6.5}$$

DEFINITION (Scott, 1980b; Barendregt, 1981). A *lambda model* is an algebra $\langle D, \cdot, K, S\rangle$ *such that*

$$K, S \in D \text{ satisfy (1.1), (1.2),}$$

$$\langle D, \cdot, \varepsilon\rangle \text{ is a combinatory model, where } \varepsilon = S(KI),$$

$$K = \varepsilon_2 \cdot K, \tag{6.6}$$

and

$$S = \varepsilon_3 \cdot S, \tag{6.7}$$

where $\varepsilon_2, \varepsilon_3$ are given by (6.2) and (6.4).

Because the righthand sides of (6.6) and (6.7) are the values of K_λ and S_λ in the environment model associated with any combinatory model $\langle D, \cdot, \varepsilon\rangle$,

they are uniquely determined independently of the particular choice of K and S.[7] Conversely, the values of K_λ and S_λ determine ε because ε is the value of $\lambda xy.\, xy$, and $\lambda xy.\, xy$ converts to $S_\lambda(K_\lambda I_\lambda)$. Thus we have established the

LAMBDA MODEL THEOREM. *Any combinatory model* $\langle D, \cdot, \varepsilon\rangle$ *uniquely determines a lambda model* $\langle D, \cdot, \varepsilon_2 \cdot K, \varepsilon_3 \cdot S\rangle$, *independently of the choice of* K *and* $S \in D$ *satisfying* (1.1–2).

Conversely, if $\langle D, \cdot, K, S\rangle$ is a lambda model, then $\langle D, \cdot, S(KI)\rangle$ is a combinatory model. Moreover, these two correspondences are inverses of each other.

In lambda models the standard algebraic notion of a *substructure* relates nicely to certain syntactic properties of lambda terms. For example, the *interior* of a lambda calculus model is normally defined as the values of the pure, i.e., constant free, *closed* lambda terms (Barendregt, 1976). In terms of lambda models, the interior now has a familiar algebraic definition as the minimum subalgebra of a lambda model; this follows immediately from (6.3), (6.5), and the Combinatory Lambda Term Lemma in Section 4.

[7] In general, ε_n is chosen to be $\mathscr{Y}_{\mathscr{B}}[\lambda x_0 \ldots x_n.\, (x_0 \ldots x_n)]$. Continuing with a purely algebraic approach, we could define, following Scott (1980b),

$$\varepsilon_1 = \varepsilon \qquad \text{and} \qquad \varepsilon_{n+1} = (B \cdot \varepsilon) \cdot (B \cdot \varepsilon_n) \qquad \text{for} \quad n > 0.$$

It is easy to verify that in any combinatory model $\langle D, \cdot, \varepsilon\rangle$,

(N.1) $\qquad \varepsilon_n d_0 \ldots d_n = d_0 d_1 \ldots d_n,$

(N.2) $\qquad$ if $\forall\, e_1, \ldots, e_n \in D \cdot d_0 e_1 \ldots e_n = d_1 e_1 \ldots e_n$ then $\varepsilon_n d_0 = \varepsilon_n d_1,$

and

(N.3) . $\qquad \varepsilon(\varepsilon_n d_0) = \varepsilon_n(\varepsilon d_0) = \varepsilon d_0$

for all $d_0, \ldots, d_n \in D$, $n > 0$.

The reader might enjoy deriving an algebraic proof solely from (N.1–3) that in any combinatory model $\langle D, \cdot, \varepsilon\rangle$ there is exactly one pair of elements K and S satisfying (1.1–2), (6.6–7).

These equations suggest another axiomatization of models proposed by Scott [cf. Volken, 1978; Barendregt, 1981, Theorem 5.4.9).

DEFINITION. Let $\mathscr{S} = \langle D, \cdot, F\rangle$ be a structure where $\cdot$ is a binary operation on D and $F \subseteq D$. Let $F_0 = D$ and $F_{n+1} = \{d_0 \in F \mid d_0 \cdot d_1 \in F_n \text{ for all } d_1 \in D\}$. $\mathscr{S}$ is a *Scott Model* if, for all $n > 0$ and any $\mathscr{C}$-term u over D such that x_0 is not free in u, $\mathscr{S}$ satisfies

$$\exists\, !x_0 \in F_n \forall\, x_1 \ldots x_n \in D[x_0 \ldots x_n = u].$$

It is easy to see that if $\mathscr{S} = \langle D, \cdot, F\rangle$ is a Scott model, then $\langle D, \cdot, \varepsilon\rangle$ is a combinatory model, where ε is the unique element of F_2 such that (1.3b) is valid in $\mathscr{S}$. Conversely, if $\langle D, \cdot, \varepsilon\rangle$ is a combinatory model, then $\langle D, \cdot, F\rangle$ is a Scott model, where $F = \{\varepsilon \cdot d \mid d \in D\} = \varepsilon \cdot D$. In fact, $F_n = \varepsilon_n \cdot D$.

From a mathematical point of view, then, lambda models are a bit nicer than combinatory models because they provide a useful notion of substructure. On the other hand, combinatory models have a simpler and much more easily checkable set of axioms—which is why I have given them emphasis in this article. The Lambda Model Theorem shows that one can easily switch to whichever of the two notions of model is more convenient at any point.

It might seem that we are now ready to develop a nice theory of models using the usual algebraic notions of substructures, morphisms, etc. One serious technical impediment remains, however. Neither the class of combinatory models nor the class of lambda models is closed under the operation of taking substructures or of applying morphisms with respect to the binary operation $\cdot$!

The difficulty springs from the fact that the first order axioms (1.3b) and (1.4) are not equations. *Equationally* axiomatized structures are guaranteed to be closed under taking substructures and morphisms, but first order axiomatizable structures are not, in general (cf. Monk, 1976, Sect. 24, or any text on Universal Algebra).

In fact the combinatory algebra which is the interior of the extensional term model is not even expandable by any choice of K, S into a (not necessarily extensional) lambda model (Barendregt, letter to Meyer, Oct. 1980); see also (Plotkin, 1974). This implies among other things that *there is no purely equational definition of lambda or combinatory models*, since equationally defined classes of structures are closed under taking substructures.

Nevertheless, there is an equationally definable class of structures called lambda algebras which serve so well for interpreting lambda terms that it is tempting to identify them as the proper algebraic embodiment of lambda calculus (cf. Lambek, 1980). We consider these next.

7. Lambda Algebras

The mapping from a lambda term u to $u^{(\mathscr{E})}$ given in Section 4 suggests an obvious way to interpret a lambda term u within an arbitrary combination *algebra*—rewrite u to be a provably equivalent term consisting solely of applications of K_λ's and S_λ's, and then replace the K_λ's by the constant $K \in D$, and similarly for S_λ. But this way of assigning values to lambda terms may not be sound unless the K and S are chosen properly.

For example, K_λ converts to $S_\lambda(K_\lambda K_\lambda) I_\lambda$. But there is no guarantee that the algebra will contain K, S such that $K = S(KK) I$.

If the combinatory algebra is a combinatory *model*, then the content of the Combinatory Model Theorem (iv) of Section 4 is that there are satisfactory

constants K, S in the model. In an arbitrary combinatory algebra no K satisfying (1.1) need behave *completely* like K_λ. However, some further purely equational conditions on combinatory algebras, strictly weaker than the axioms for combinatory models, are sufficient to guarantee the existence of well behaved K and S. These conditions are embodied in the following axioms.

DEFINITION (Curry; cf. Barendregt, 1981, Chap. 7). A *lambda algebra* is a structure $\langle D, \cdot, K, S\rangle$, where $\langle D, \cdot\rangle$ is a combinatory algebra, $K, S \in D$ satisfy (1.1), (1.2), and

$$K = \langle x\rangle(\langle y\rangle(Kxy)), \tag{7.1}$$

$$S = \langle x\rangle(\langle y\rangle(\langle z\rangle(Sxyz))), \tag{7.2}$$

$$\langle x\rangle(\langle y\rangle(S(Kx)(Ky))) = \langle x\rangle(\langle y\rangle(K(xy))), \tag{7.3}$$

$$\langle x\rangle(\langle y\rangle(S(S(KK)\,x)\,y)) = \langle x\rangle(\langle y\rangle(\langle z\rangle(xz))), \tag{7.4}$$

$$\langle x\rangle(\langle y\rangle(\langle z\rangle(S(S(S(KS)\,x)\,y)\,z) = \langle x\rangle(\langle y\rangle(\langle z\rangle(S(Sxz)(Syz)). \tag{7.5}$$

Note that (7.1–5) denote equations between constants. For example, (7.1) in less abbreviated form reads

$$K = S(S(KS)(S(KK)(S(KK)\,I)))(KI)$$

which would be even longer if we had expanded the combinator I as SKK and put in full parenthesization. The reader will appreciate the utility of the abbreviations.[8] Even with the abbreviations, (7.1–5) are hardly memorable, having been chosen solely for the purpose of carrying out the proofs below.

We now prove the rather surprising fact that combinatory models can be obtained from lambda algebras simply by extending lambda algebras with indeterminates—just as the domain of integer multivariate polynomials is obtained from the ring of integers.[9] This result will then yield another, mathematically robust, characterization of lambda algebras which is surely not apparent from their definition.

[8] Our definition of the transformation $\langle x\rangle$ on $\mathscr{C}$-terms was chosen for ease in proofs rather than efficiency, and consequently the length of $\langle x\rangle\, u$ has been allowed to grow exponentially in the length of u. There exist transforms with the same properties as $\langle x\rangle$ which increase the length only linearly (Turner, 1979).

[9] Barendregt and Koymans (1980) show that not all combinatory algebras can be *expanded* by choice of K and S (as opposed to *extended* by the addition of new elements possibly including K and S) into lambda algebras. The interior of the combinatory word algebra based on K, S-terms is an example of such a combinatory algebra. They also show, as noted at the end of Section 6, that not all lambda algebras are lambda models.

Let $\mathscr{C} = \langle D, \cdot \rangle$ be a combinatory algebra, and $K, S \in D$ be any elements satisfying (1.1–2).

Let X be a set of variables and $\mathscr{C}[X]$ be the free combinatory algebra generated by X over the constants in $\mathscr{C}$. That is, $\mathscr{C}[X]$ is the free word algebra of $\mathscr{C}$-terms with variables only from X, modulo the congruence relation on $\mathscr{C}$-terms generated by the equations between constant terms valid in $\mathscr{C}$ and all substitution instances of (1.1–2).

Formally, let u, v, w range over $\mathscr{C}$-terms, and define the *proof system of β-Combinatory Logic* for $\mathscr{C}$ to have axioms

$$u = v \text{ such that } u, v \text{ are variable free terms and } \mathscr{C} \models u = v,$$

$$Kuv = u,$$

and

$$Suvw = uw(vw),$$

and inference rules: (transitivity and symmetry) and (congruence). (Again, we would also insist on the axiom (reflexivity) except that it follows already from $Kuv = u$ and (transitivity and symmetry).)

Write $\mathscr{C}\text{-}CL_\beta \vdash u = v$ iff equation $u = v$ is provable in this system, and let

$$[\![u]\!] = \{v \mid \mathscr{C}\text{-}CL_\beta \vdash u = v\},$$

$$D[X] = \{[\![u]\!] \mid u \in \Lambda(D) \text{ and all variables in } u \text{ are in } X\}.$$

Then

$$\mathscr{C}[X] = \langle D[X], \cdot \rangle, \qquad \text{where } [\![u]\!] \cdot [\![v]\!] = [\![(uv)]\!].$$

(7.6) LEMMA. *$\mathscr{C}[X]$ is a combinatory algebra and the mapping taking $d \in D$ to $[\![d]\!]$ isomorphically embeds $\mathscr{C}$ into $\mathscr{C}[X]$. Moreover, if u, v are $\mathscr{C}$-terms all of whose variables are in the set X, then*

$$\mathscr{C}[X] \models u = v \qquad \text{iff} \quad \mathscr{C}\text{-}CL_\beta \vdash u = v.$$

Proof. The construction of $\mathscr{C}[X]$ from $\mathscr{C}$ is the standard one for constructing a "polynomial" algebra from any equationally defined algebra. ∎

Lemma (7.6) justifies identifying $d \in D$ with the element $[\![d]\!]$ of $D[X]$ which we shall continue to do. Note that because (7.1–5) denote equations between variable-free $\mathscr{C}$-terms, it now follows immediately that $\mathscr{C}[X]$ satisfies whichever of (7.1–5) that $\mathscr{C}$ satisfies. In particular, if $\mathscr{C}$ is a lambda algebra, then so is $\mathscr{C}[X]$.

In the Appendix we demonstrate via Lemmas (7.7–11) that Eqs. (7.1–5) imply the following key technical property about the $\langle x \rangle$-transformation of $\mathscr{C}$-terms.

(7.12) LEMMA. *Let $\mathscr{C}$ be a lambda algebra and u, v be $\mathscr{C}$-terms with variables only in X. If $\mathscr{C}[X] \vDash u = v$, then $\mathscr{C}[X] \vDash \langle x \rangle u = \langle x \rangle v$.*

A point of possible confusion about (7.12) is that it does not hold in $\mathscr{C}$ as opposed to $\mathscr{C}[X]$. That is, it may be that the equation $u = v$ is valid in $\mathscr{C}$, but the equation $\langle x \rangle u = \langle x \rangle v$ is not. The source of the confusion is that while $\mathscr{C}[X] \vDash u = v$ implies $\mathscr{C} \vDash u = v$, the converse fails. (This frequently happens in classical algebras. For example, $x = x^2$ is valid in the ring $\mathbb{Z}_2$, but not in the polynomial ring $\mathbb{Z}_2[x]$.) The key property of $\mathscr{C}[X]$ required in the proof of (7.12) is the equivalence of validity and provability given by (7.6) which holds only for $\mathscr{C}$-terms all of whose variables are in X.

We can now state precisely the relation between lambda algebras and lambda models.

LAMBDA ALGEBRA THEOREM. (i) *If $\mathscr{C} = \langle D, \cdot, K, S \rangle$ is a lambda algebra and X is an infinite set of variables, then $\mathscr{C}_\lambda[X] = \langle D[X], \cdot, K, S \rangle$ is a lambda model.*

(ii) *Conversely, every lambda model is a lambda algebra.*

So given a lambda algebra $\mathscr{C}$, we can always extend it with at most a countable number of indeterminates to obtain a lambda model.

COROLLARY (Barendregt). (i) *The lambda algebras are precisely the class of all substructures of lambda models.*

(ii) *The lambda algebras are precisely the class of all homomorphic images of lambda models.*

Proof. Applying homomorphisms and taking substructures preserves equations, so homomorphic images and substructures of models are algebras; i.e., the set of lambda algebras contains the images and substructures of the lambda models. The reverse containment follows because every lambda algebra $\mathscr{C}$ is an image and a substructure of $\mathscr{C}_\lambda[X]$. ∎

Thus, we learn the unexpected facts that the class of homomorphic images and the class of substructures of lambda models *coincide* and are *finitely axiomatizable* by equations, namely the axioms for lambda algebras. This is the robust characterization of lambda algebras promised above.

Extensional combinatory algebras and extensional combinatory models coincide. To characterize their substructures axiomatically, just add the axiom $I = \langle x \rangle(\langle y \rangle(xy))$ to (7.1–5). The resulting class of algebras are called *Curry algebras* (cf. Lambek, 1980).

Clearly this connection between lambda algebras and lambda models is very intimate. It is esentially the same as the relation between a ring of coefficients and its polynomial domain. Several researchers have preferred to

give the coefficients priority over the polynomials as it were, and propose to define "models" of the lambda calculus to be lambda algebras. The principal argument in support of this view is the feeling that the values of the closed terms of lambda calculus ought to be a model and, more generally, that models ought to be closed under $\cdot$-morphisms and taking substructures—properties possessed by lambda algebras but not by lambda models.

In contrast, my view is that soundness of rule (ξ) is essential. It embodies the idea that $\lambda x \,.\, u$ defines a function which is determined uniquely by the value of the term u in each possible environment *over the model.* Rule (ξ) is sound for lambda models, but *not* for lambda algebras, as we noted following (7.12).

Of course as long as all concerned are aware of the distinction between the polynomial domain and coefficient ring, it hardly matters which is given priority, but ignoring the distinction has been a source of some confusion.

The same distinction in another guise arises between the algebraic formulation of a system of functional types embodied in the axioms for a Cartesian Closed Category (CCC) and a *concrete CCC*, which actually is a system of functions of higher types. The connections between lambda calculus and CCCs has been emphasized by several authors (cf. Lambek, 1980; Scott, 1980b; Obtulowicz and Wiweger, 1978; Koymans, 1981). In particular, Koymans (1981) establishes a natural isomorphism between CCCs and lambda algebras, and between concrete CCCs and lambda models. Thus the equational axioms for CCCs provide a mathematically meaningful axiomatization of lambda algebras, a notable improvement over (7.1–5). That, however, is another story.

8. Summary and Further Directions

By mimicking how ordinary terms are evaluated over an algebraic structure, we developed the idea of evaluating a lambda term over a functional domain. This led directly to the formulation of environment models. The Soundness and Completeness Theorems of Section 2 confirmed that environment models precisely captured and *justified* the informal intuition behind the classical calculus of lambda terms.

The development in subsequent sections revealed how to treat lambda calculus as a theory of equations for a class of ordinary algebraic structures. In particular, the Combinatory Model Theorem of Section 4 described a bijection between environment models and combinatorial models, and showed how to translate effectively between lambda terms and equivalent combinatory terms. The Lambda Model Theorem of Section 6 described a similar bijection between combinatory models and lambda models. In fact, the Soundness and Completeness Theorems can also be understood as

describing natural correspondences between environment models and lambda theories.[10] So

1. environment models,
2. lambda theories,
3. combinatory models,
4. lambda models

can be regarded as alternative formulations of the same concept. Finally, by considering substructures and images of models, we developed in Section 7 the class of

5. lambda algebras.

Lambda algebras have a finite equational axiomatization, and therefore are closed under forming substructures and images. They properly include the class of models, but determine models in precisely the same way that the ring of coefficients determines the the corresponding domain of multivariate polynomials.

Algebraic definitions and arguments can often offer more simplicity and greater appeal than syntactic ones, particularly if one can avoid the notorious pitfalls of substitution in the presence of bound variables. Having in principle eliminated the need for syntactic notions in defining which structures are models, the general question arises of how much more of the highly developed syntactic–computational "proof theory" of lambda calculus can be usefully understood from an algebraic "model theory" viewpoint. There has already been valuable interaction between the two viewpoints. One important example is worth sketching.

A lambda term has a *head normal form* if it converts to a term of the form $\lambda x_1 \ldots x_n \,.\, (yu)$ for some $n \geqslant 0$; "$\lambda x_1 \ldots x_n \,.\, y$" is called the *head* of the term and is unique up to renaming bound variables (for η-calculus there is a slightly more complicated kind of uniqueness property). A lambda term is *unsolvable* if it does not have a head normal form. By repeatedly converting the solvable subterms of any term u into head normal form and replacing unsolvable subterms by a new constant Ω, one obtains in the limit a unique, possibly infinite, term called the *Bohm tree* of u. The Bohm tree can be regarded as the trace of the possibly infinite computation needed to evaluate the term. Following earlier work in (Hyland, 1976; Wadsworth, 1976;

[10] The lambda theory of the term model constructed from a given lambda theory $\mathscr{E}$ as in the proof of the Completeness Theorem is a *conservative extension* of $\mathscr{E}$, but not the same as $\mathscr{E}$. So the correspondences established in Section 2 in each direction between lambda theories and environment models are not quite inverses. A natural bijection could be establised between models and theories *with enough constants*, viz., such that every closed term is provably equivalent to a constant.

Plotkin, 1978; Barendregt and Longo, 1980), Longo has recently observed that the value in D_A of a closed term u is set theoretically included in the value of a closed term v iff the Bohm tree of u *approximates* that of v; namely, the Bohm tree of u is obtainable from the Bohm tree of v by replacing some of the subterms of the tree of v by Ω [Longo, 1981]. In particular, an equation between lambda terms is valid in D_A iff the terms have the same Bohm tree. This provides an elegant connection between the syntactic–computational behavior of lambda terms and their meaning in a mathematically elementary model.[11]

As Scott (1980b) has emphasized, the *un*typed lambda calculus considered above can be viewed as the special case of the typed lambda calculus in which there is a "universal" type into which all other types can be isomorphically embedded. Most applications of lambda calculus in the study of programming languages and computability require the richer structure of multiple types (cf. Stoy, 1977; Gordon, 1979). I hope to provide an elementary treatment of this generalization in a sequel tentatively titled "What is a solution of a domain equation?"

APPENDIX: PROOFS

From Section 3

SOUNDNESS THEOREM. *The equations valid in an environment model form a lambda theory. In particular, if u converts to v, then $u = v$ is valid in all environment models.*

Proof. By (3.5) and (3.6), $\mathscr{V}[(uv)]$ and $\mathscr{V}[(\lambda x\, u)]$ are determined solely by $\mathscr{V}[u]$ and $\mathscr{V}[v]$, so (congruence) and (ξ) preserve validity.

To verify that (α) is valid, let $\mathscr{V}[(\lambda x\, u)]\rho = \Psi(f)$ as in (3.6), so that $f(d) = \mathscr{V}[u](\rho\{d/x\})$. Let y be a variable distinct from x and such that y is not free in u. Let $\mathscr{V}[(\lambda y\, u[y/x])]\rho = \Psi(g)$, so $g(d) = \mathscr{V}[u[y/x]](\rho\{d/y\})$. Then $\lambda x\,.\,u = \lambda y\,.\,u[y/x]$ will be valid providing $f = g$.

By the Substitution Lemma, $g(d) = \mathscr{V}[u]((\rho\{d/y\})\{d'/x\})$, where $d' = \mathscr{V}[y](\rho\{d/y\})$. By (3.2), $d' = d$. Also, $(\rho\{d/y\})\{d'/x\} = (\rho\{d'/x\})\{d/y\}$ by definition since $y \neq x$, so $g(d) = \mathscr{V}[u]((\rho\{d/x\})\{d/y\})$. By the Substitution Lemma again, $g(d) = \mathscr{V}[u[x/y]](\rho\{d/x\})$, but since y is not free in u, $u[x/y] = u$, so $g(d) = f(d)$.

Verification of (β) follows even more easily from the Substitution Lemma.

[11] It also provides a simple model theoretic characterization of the syntactic concept of *normal form*, as pointed out by Longo. Namely, $d \in D_A$ is the value of a pure closed lambda term in normal form iff d is maximal under set inclusion in the interior of D_A and contains only finitely many elements of the interior.

Finally, if u converts to v, then the equation $u = v$ is in every lambda theory, and hence is in the lambda theory of equations valid in any particular environment model. ∎

COMPLETENESS THEOREM. *Every lambda theory consists of precisely the equations valid in some environment model. That is, for every lambda theory* $\mathscr{T}$, *there is an environment model* $\mathscr{E}$ *such that*

$$\vdash_{\mathscr{T}} u = v \qquad \text{iff} \quad \mathscr{E} \models u = v.$$

In particular, $u = v$ is valid in all environment models iff u converts to v.

Proof. Let $\mathscr{T}$ be a lambda theory and $\mathscr{E}$ its associated term model. First we verify that the $\mathscr{E}$ is a functional domain.

Note that the (congruence) rule implies that Φ in the definition of term model is well defined. To see that Ψ is well defined, note that if $\Phi([\![u]\!]_{\mathscr{T}}) = \Phi([\![v]\!]_{\mathscr{T}})$, then evaluating at argument $[\![x]\!]_{\mathscr{T}}$ for x not free in u, v yields $[\![ux]\!]_{\mathscr{T}} = [\![vx]\!]_{\mathscr{T}}$, so (ξ) implies $[\![\lambda x \,.\, ux]\!]_{\mathscr{T}} = [\![\lambda x \,.\, vx]\!]_{\mathscr{T}}$; that is, $\Psi(\Phi([\![u]\!]_{\mathscr{T}})) = \Psi(\Phi([\![v]\!]_{\mathscr{T}}))$ by definition. Finally, (β) immediately implies that Φ is a left inverse of Ψ, so $\langle D, \Phi, \Psi\rangle$ is a functional domain.

Next, we verify that $\mathscr{E}$ is indeed an environment model.

For any environment ρ, let $u[\rho]$ abbreviate $[\![u[u_1/x_1, \ldots, u_n/x_n]\,]\!]_{\mathscr{T}}$, where $x_1, \ldots, x_n$ are the free variables of u and $\rho(x_i) = [\![u_i]\!]_{\mathscr{T}}$. Note that $u[\rho]$ is well defined since simultaneous substitution preserves equations.

We claim that for all $u \in \Lambda(C)$ and environments ρ, if $\mathscr{V}_{\mathscr{E}}[u]\,\rho$ is defined by (3.1–2), (3.5–6), then

$$\mathscr{V}_{\mathscr{E}}[u]\,\rho = u[\rho].$$

In particular, $\mathscr{E}$ is an environment model. (Alert readers may remember that our formal definitions require that constants in lambda terms must be elements of D, so we must identify constants $c \in C$ with the corresponding constants $[\![c]\!]_{\mathscr{T}} \in D$.)

The claim follows by induction on the definition of a lambda term u. We consider only the most difficult case when u is of the form $\lambda x \,.\, v$, where x is free in v. In this case,

$$\mathscr{V}_{\mathscr{E}}[u]\,\rho = \Psi(\lambda d \in D \,.\, \mathscr{V}_{\mathscr{E}}[v](\rho\{d/x\}))$$

by (3.6), providing the argument of Ψ is in the range of Φ. So it suffices to prove that $\lambda d \in D \,.\, \mathscr{V}_{\mathscr{E}}[v](\rho\{d/x\}) = \Phi((\lambda \,.\, v)[\rho])$.

But for any $[\![w]\!]_{\mathscr{E}} \in D$,

$$\begin{aligned}(\lambda d \in D \cdot \mathscr{V}_{\mathscr{E}}[v](\rho\{d/x\}))([\![w]\!]_{\mathscr{E}}) &= \mathscr{V}_{\mathscr{E}}[v](\rho\{[\![w]\!]_{\mathscr{E}}/x\}) \\ &= [\![v[u_1/x_1,\ldots,u_n/x_n, w/x]\,]\!]_{\mathscr{E}} \quad \text{by induction hypothesis.}\end{aligned}$$

Now let y, z be a new variables not free in v, w, $u_1,\ldots,u_n$, and let $u_i' = u_i[z/x]$. Then by the definition of simultaneous substitution,

$$\begin{aligned}[\![v[u_1/x_1,\ldots,u_n/x_n, w/x]\,]\!]_{\mathscr{E}} &= [\![v[u_1'/x_1,\ldots,u_n'/x_n][w/x][x/z]\,]\!]_{\mathscr{E}} \\ &= [\![((\lambda x\,.\,(v[u_1'/x_1,\ldots,u_n'/x_n]))\,w)[x/z]\,]\!]_{\mathscr{E}} \quad \text{by } (\beta) \\ &= [\![((\lambda x\,.\,(v[u_1'/x_1,\ldots,u_n'/x_n]))[x/z]\,w)]\!]_{\mathscr{E}} \quad \text{since } z \text{ is not free in } w \\ &= [\![(\lambda y\,.\,(v[u_1'/x_1,\ldots,u_n'/x_n][y/x]))[x/z]\,w)]\!]_{\mathscr{E}} \quad \text{renaming } \lambda x \text{ to } \lambda y \text{ to avoid capture of } x \\ &= [\![((\lambda y\,.\,(v[y/x][u_1'/x_1,\ldots,u_n'/x_n]))[x/z]\,w)]\!]_{\mathscr{E}} \quad \text{since } x \text{ is not free in } u_i' \\ &= [\![(\lambda y\,.\,(v[y/x][u_1/x_1,\ldots,u_n/x_n])\,w)]\!]_{\mathscr{E}} \quad \text{by definition of } u_i' \text{ since } z \text{ is not free in } v[y/x] \\ &= [\![((\lambda y\,.\,(v[y/x]))[u_1/x_1,\ldots,u_n/x_n]\,w)]\!]_{\mathscr{E}} \quad \text{since } y \text{ is not free in } u_i \\ &= [\![(\lambda x\,.\,v)[u_1/x_1,\ldots,u_n/x_n]\,w)]\!]_{\mathscr{E}} \quad \text{renaming } \lambda y \text{ to } \lambda x \text{ by } (\alpha) \\ &= (\Phi((\lambda x\,.\,v)[\rho]))([\![w]\!]_{\mathscr{E}}) \quad \text{by definition of } [\rho] \text{ and } \Phi.\end{aligned}$$

Therefore, $\lambda d \in D\,.\,\mathscr{V}_{\mathscr{E}}[v](\rho\{d/x\}) = \Phi((\lambda x\,.\,v)[\rho])$, and the claim is proved.

Now if $\mathscr{E} \models u = v$, then in the particular environment ρ_0 such that $\rho_0(x) = [\![x]\!]_{\mathscr{E}}$ for all variables x, the terms u and v have the same value. By the above claim, the value of u is $[\![u]\!]_{\mathscr{E}}$ and that of v is $[\![v]\!]_{\mathscr{E}}$, so $[\![u]\!]_{\mathscr{E}} = [\![v]\!]_{\mathscr{E}}$. That is, $\vdash_{\mathscr{E}} u = v$.

Conversely, if $\vdash_{\mathscr{E}} u = v$, then $\vdash_{\mathscr{E}} u[\rho] = v[\rho]$ for all ρ since simultaneous substitution preserves equations. The above claim immediately implies that $\mathscr{E} \models u = v$. ∎

From Section 4

COMBINATORY MODEL THEOREM. (i) *The functional domain* $\mathscr{E}$ *associated with a combinatory model* $\mathscr{C}$ *is an environment model which assigns the same values to* $\mathscr{C}$*-terms. That is,*

$$\mathscr{V}_{\mathscr{E}}[u] = \mathscr{V}_{\mathscr{E}}[u]$$

for all $\mathscr{C}$*-terms* u.

(ii) *The algebra associated with an environment model is a combinatory model.*

(iii) *The associations between combinatory models and environment models defined above are inverses of each other. That is, if $\mathscr{E}$ is the environment model associated with a combinatory model $\mathscr{C}$, then $\mathscr{C}$ is the combinatory model associated with $\mathscr{E}$, and vice versa.*

Proof. (i) Let $\bar{x}_n$ abbreviate the sequence of distinct variables $x_1,..., x_n$. Let $\bar{d}_n$ abbreviate the sequence $d_1,..., d_n$ of (not necessarily distinct) elements in D, and let $\rho\{\bar{d}_n/\bar{x}_n\}$ abbreviate $(\ldots(\rho\{d_1/x_1\})\ldots\{d_n/x_n\})$. We claim that for every term $u \in \Lambda(D)$, for every environment ρ, and all $\bar{x}_n$, there is an element $d_{u\rho\bar{x}_n} \in D$ such that

$$\mathscr{V}_{\mathscr{E}}[u](\rho\{\bar{d}_n/\bar{x}_n\}) = d_{u\rho\bar{x}_n}\bar{d}_n$$

for all $\bar{d}_n \in D$.

This claim follows by induction on the definition of lambda terms. We give the details only for the most difficult case that u is of the form $\lambda x_{n+1} \cdot v$, where x_{n+1} is distinct from $\bar{x}_n$. We have

$$\begin{aligned}
&\mathscr{V}_{\mathscr{E}}[\lambda x_{n+1} \cdot v](\rho\{\bar{d}_n/\bar{x}_n\}) \\
&\quad = \Psi(\lambda d_{n+1} \in D \cdot \mathscr{V}_{\mathscr{E}}[v](\rho\{\bar{d}_n/\bar{x}_n\})\{d_{n+1}/x_{n+1}\})) \qquad \text{by (3.6)} \\
&\quad = \Psi(\lambda d_{n+1} \in D \cdot \mathscr{V}_{\mathscr{E}}[v](\rho\{\bar{d}_{n+1}/\bar{x}_{n+1}\})) \\
&\quad = \Psi(\lambda d_{n+1} \in D \cdot d_{v\rho\bar{x}_{n+1}}\bar{d}_{n+1}) \quad \text{by induction hypothesis} \\
&\quad = (\varepsilon(d_{v\rho\bar{x}_{n+1}}\bar{d}_n)) \qquad \text{by definition of } \Psi.
\end{aligned}$$

By combinatory completeness, there is a $d \in D$ such that

$$\mathscr{C} \vDash (\varepsilon(d_{v\rho\bar{x}_{n+1}}\bar{x}_n)) = d\bar{x}_n,$$

so we define $d_{u\rho\bar{x}_n}$ to be d.

The claim immediately implies that $\lambda d \in D \cdot \mathscr{V}_{\mathscr{E}}[u](\rho\{d/x\}) = \Phi(d_{u\rho x}) \in D \to D$, so that the functional domain $\mathscr{E}$ is an environment model.

Induction on the definition of $\mathscr{C}$-terms establishes that $\mathscr{V}_{\mathscr{E}}$ and $\mathscr{V}_{\mathscr{C}}$ coincide on $\mathscr{C}$-terms.

(ii) Let $\mathscr{E} = \langle D, \Phi, \Psi\rangle$ be an environment model and $\mathscr{C}$ its associated algebra. Choose K, $S \in D$ to be $\mathscr{V}_{\mathscr{E}}[\lambda xy \cdot x]$ and $\mathscr{V}_{\mathscr{E}}[\lambda xyz \cdot xz(yz)]$, respectively. Then (1.1), (1.2), (1.3a, c) follow directly from the Soundness Theorem, (β), and the definitions. To verify (1.3b), note that

$$\begin{aligned}
\varepsilon \cdot d_0 &= (\Phi(\mathscr{V}_{\mathscr{E}}[\lambda xy \cdot xy]))(d_0) \qquad \text{by definition of } \Phi \text{ and } \varepsilon \\
&= (\lambda d \in D \cdot \mathscr{V}_{\mathscr{E}}[\lambda y \cdot xy](\rho\{d/x\}))(d_0) \qquad \text{by (3.4) and (3.6)}
\end{aligned}$$

$$= \mathscr{V}_{\mathscr{E}}[\lambda y \,.\, xy](\rho\{d_0/x\})$$
$$= \Psi(\lambda d \in D \,.\, \mathscr{V}_{\mathscr{E}}[(xy)]((\rho\{d_0/x\})\{d/y\})) \qquad \text{by (3.6)}$$
$$= \Psi(\lambda d \in D \,.\, (\Phi(d_0))(d)) \qquad \text{by (3.2), (3.4), (3.5)}$$
$$= \Psi(\Phi(d_0)).$$

Hence, if $d_0 \cdot d = d_1 \cdot d$ for all d, then $\Phi(d_0) = \Phi(d_1)$ by definition of Φ, so $\varepsilon \cdot d_0 = \Psi(\Phi(d_0)) = \Psi(\Phi(d_1)) = \varepsilon \cdot d_1$. This proves (1.3b) holds in $\mathscr{C}$.

(iii) Let $\mathscr{E} = \langle D, \Phi, \Psi \rangle$ be an environment model, $\mathscr{C} = \langle D, \cdot, \varepsilon \rangle$ the associated combinatory model, $\mathscr{E}' = \langle D, \Phi', \Psi' \rangle$ the environment model associated with $\mathscr{C}$, and $\mathscr{C}'$ the combinatory model associated with $\mathscr{E}'$, Clearly, $\Phi = \Phi'$. But $\Psi'(\Phi(d_0)) = \varepsilon \cdot d_0$ by definition of Ψ', and $\varepsilon \cdot d_0 = \Psi(\Phi(d_0))$ by the proof of (ii) above, so $\Psi = \Psi'$. Hence, $\mathscr{E} = \mathscr{E}'$. The proof that $\mathscr{C} = \mathscr{C}'$ follows similarly. ∎

From Section 7

(7.7) LEMMA. *For any $\mathscr{C}$-term u and variables x, y,*

(i) *x does not occur in $\langle x \rangle u$,*

(ii) *if y does not occur in u, then $\langle x \rangle u = \langle y \rangle (u[y/x])$.*

Proof. By induction on the definition of $\langle x \rangle$. ∎

(7.8) LEMMA. *For all $\mathscr{C}$-terms u, v and distinct variables x, y, if x does not occur in v, then*

$$(\langle x \rangle u)[v/y] = \langle x \rangle (u[v/y]).$$

Proof. By induction on the definition of $\langle x \rangle$. The cases that $u = x$ or x does not occur in u are trivial.

Suppose $u = (u_1 u_2)$ and x occurs in u. Then

$$(\langle x \rangle u)[v/y] = (S(\langle x \rangle u_1)(\langle x \rangle u_2))[v/y] \qquad \text{by definition of } \langle x \rangle$$
$$= S((\langle x \rangle u_1)[v/y])((\langle x \rangle u_2)[v/y]) \qquad \text{by definition of } [v/y]$$
$$= S(\langle x \rangle (u_1[v/y]))(\langle x \rangle (u_2[v/y])) \qquad \text{by induction hypothesis}$$
$$= \langle x \rangle (u_1[v/y]\, u_2[v/y]) \qquad \text{by definition of } \langle x \rangle$$
$$= \langle x \rangle (u[v/y]) \qquad \text{by definition of } [v/y]. \quad ∎$$

(7.9) LEMMA. *For any $\mathscr{C}$-terms u, v and variable x, $\mathscr{C} \models ((\langle x \rangle u)\, v) = u[v/x]$.*

Proof. As already observed in the Combinatory Completeness Lemma, an induction on the definition of $\langle x\rangle$ implies $\mathscr{C} \models (\langle x\rangle u)\, x = u$. Since substitution preserves validity of equations, $\mathscr{C} \models ((\langle x\rangle u)\, x)[v/x] = u[v/x]$, but by (7.7(i)), x does not occur in $\langle x\rangle u$, so $((\langle x\rangle u)\, x)[v/x] = ((\langle x\rangle u)\, v)$. ∎

(7.10) LEMMA. *If K, S satisfy* (7.1–2), *then for all distinct variables x, y, z, and $\mathscr{C}$-terms u*

(i) $\mathscr{C} \models \langle y\rangle(Kxy) = Kx$,

(ii) $\mathscr{C} \models \langle z\rangle(Sxyz) = Sxy$,

(iii) $\mathscr{C} \models \langle y\rangle((\langle x\rangle u)\, y) = \langle x\rangle u$ *if y does not occur in* $\langle x\rangle u$.

Proof. (i) $\mathscr{C} \models \langle y\rangle(Kxy) = (\langle x\rangle(\langle y\rangle(Kxy)))\, x$ by (7.9)

$$= Kx \qquad \text{by (7.1).}$$

(ii) $\mathscr{C} \models \langle z\rangle(Sxyz) = (\langle x\rangle(\langle y\rangle(\langle z\rangle(Sxyz))))\, xy$ by (7.9) twice

$$= Sxy \qquad \text{by (7.2).}$$

(iii) By definition $\langle x\rangle u$ is always of the form Kv or Svw. In the first case, $\langle x\rangle u = Kv = (Kx)[v/x]$, but

$$\begin{aligned}\mathscr{C} \models (Kx)[v/x] &= (\langle y\rangle(Kxy))[v/x] \qquad \text{by (i),}\\ &= \langle y\rangle((Kxy)[v/x]) \qquad \text{by (7.8) providing } y \text{ does not occur in } v\\ &= \langle y\rangle(Kvy) = \langle y\rangle((\langle x\rangle u)\, y).\end{aligned}$$

The case $u = Svw$ follows similarly from (ii). ∎

(7.11) LEMMA. *If K, S satisfy* (7.3), *then for all $\mathscr{C}$-terms u, v and variables x,*

$$\mathscr{C} \models \langle x\rangle(uv) = S(\langle x\rangle u)(\langle x\rangle v).$$

Proof. If x occurs in (uv), then the equation is identically true, so assume x does *not* occur in (uv). Let y, z be distinct variables not equal to x and not occurring in (uv). Then

$$\begin{aligned}\langle x\rangle(uv) &= K(uv) \qquad \text{by definition of } \langle x\rangle\\ &= K(uy)[v/y] \qquad \text{since } y \text{ does not occur in } u.\end{aligned}$$

But

$$\begin{aligned}
\mathscr{C} \vDash K(uy)[v/y] &= (\langle y\rangle(K(uy))\, v) \qquad \text{by (7.9)}\\
&= (\langle y\rangle((K(xy))[u/x])\, v)\\
&= ((\langle y\rangle(K(xy)))[u/x]\, v) \qquad \text{by (7.8) since } y \text{ does not occur in } u\\
&= (\langle x\rangle(\langle y\rangle(K(xy))))\, uv \qquad \text{by (7.9)}\\
&= (\langle x\rangle(\langle y\rangle(S(Kx)(Ky))))\, uv \qquad \text{by (7.3)}\\
&= ((\langle y\rangle(S(Kx)(Ky)))[u/x]\, v) \qquad \text{by (7.9)}\\
&= (\langle y\rangle((S(Kx)(Ky))[u/x])\, v) \qquad \text{by (7.8) since } y \text{ does not occur in } u\\
&= (\langle y\rangle(S(Ku)(Ky))\, v) \qquad \text{by substitution}\\
&= (S(Ku)(Ky))[v/y] \qquad \text{by (7.9)}\\
&= S(Ku)(Kv) \qquad \text{since } y \text{ does not occur in } u\\
&= S(\langle x\rangle u)(\langle x\rangle v) \qquad \text{by definition of } \langle x\rangle. \ \blacksquare
\end{aligned}$$

(7.12) LEMMA. *Let $\mathscr{C}$ be a lambda algebra and u, v be $\mathscr{C}$-terms with variables only in X. If $\mathscr{C}[X] \vDash u = v$, then $\mathscr{C}[X] \vDash \langle x\rangle u = \langle x\rangle v$.*

Proof. By (7.6) validity is the same as provability for equations between $\mathscr{C}$-terms u, v all of whose variables are in X. We proceed by induction on the length of the proof that $\mathscr{C}\text{-}CL_\beta \vdash u = v$.

If the proof is of length one, i.e., $u = v$ is an axiom, then if u, v are variable free terms, the result is immediate. If $u = Ku_1u_2$ and $v = u_1$, then, noting that (7.11) holds for $\mathscr{C}[X]$ because, by (7.6), $\mathscr{C}[X]$ is a combinatory algebra satisfying the same variable free equations as $\mathscr{C}$, we have $\langle x\rangle u = \langle x\rangle v$ because

$$\begin{aligned}
\mathscr{C}[X] \vDash \langle x\rangle(Ku_1u_2) &= S(\langle x\rangle(Ku_1))(\langle x\rangle u_2) \qquad \text{by (7.11)}\\
&= S(S(KK)(\langle x\rangle u_1))(\langle x\rangle u_2) \qquad \text{by (7.11)}\\
&= (S(S(KK)(\langle x\rangle u_1))\, y)[\langle x\rangle u_2/y] \qquad \text{where } y \text{ is chosen not to occur in } \langle x\rangle u_1\\
&= (\langle y\rangle(S(S(KK)(x)\, u_1))\, y))\langle x\rangle u_2 \qquad \text{by (7.9)}\\
&= (\langle y\rangle((S(S(KK)\, x)\, y)[\langle x\rangle u_1/x]))\langle x\rangle u_2\\
&= ((\langle y\rangle((S(S(KK)\, x)\, y)))[\langle x\rangle u_1/x])\langle x\rangle u_2 \qquad \text{by (7.8) since } y \text{ is not in } \langle x\rangle u_1
\end{aligned}$$

$= (\langle x\rangle(\langle y\rangle((S(S(KK)\,x)\,y))))\langle x\rangle\, u_1\langle x\rangle\, u_2$ by (7.9)

$= (\langle x\rangle(\langle y\rangle(\langle z\rangle(xz))))\langle x\rangle\, u_1\langle x\rangle\, u_2$ by (7.4)

$= \langle z\rangle((\langle x\rangle\, u_1)\, z)$ by (7.8–9), where z is chosen not to occur in $\langle x\rangle\, u_1$,

$= \langle x\rangle\, u_1$ by (7.10(iii)).

The case that $u = v$ is the axiom $Su_1u_2u_3 = u_1u_3(u_2u_3)$ follows similarly using (7.5). So (7.12) holds for the axioms of $\mathscr{C}$-CL_β.

If the last inference rule in the proof of $u = v$ was (transitivity and symmetry), then (7.12) follows immediately by induction. If the last rule was (congruence), then $u = (u_1u_2)$, $v = (v_1v_2)$, and $\mathscr{C}[X] \models u_1 = v_1$, $u_2 = v_2$. Hence,

$\mathscr{C}[X] \models \langle x\rangle\, u = S(\langle x\rangle\, u_1)(\langle x\rangle\, u_2)$ by (7.11)

$= S(\langle x\rangle\, v_1)(\langle x\rangle\, v_2)$ by induction since substitution preserves validity

$= \langle x\rangle\, v$ by (7.11). ∎

LAMBDA ALGEBRA THEOREM. (i) *If $\mathscr{C} = \langle D, \cdot, K, S\rangle$ is a lambda algebra and X is an infinite set of variables, then $\mathscr{C}_\lambda[X] = \langle D[X], \cdot, K, S\rangle$ is a lambda model.*

(ii) *Conversely, every lambda model is a lambda algebra.*

Proof. (i) Let $\varepsilon = \langle x\rangle(\langle y\rangle(xy))$. We first observe that $\langle D[X], \cdot, \varepsilon\rangle$ is a combinatory model. To see this, note that by (7.8–9), $\varepsilon d = \langle y\rangle(dy)$ for all $d \in D[X]$. Eq. (1.3a) follows directly by another application of (7.9), and (1.3c) follows by (7.10(iii)).

To verify (1.3b), suppose $[\![u]\!]\, d = [\![v]\!]\, d$ for all d in $D[X]$. Let $y \in X$ be a variable not in u, v; there is such a y since X is infinite. Then letting d be $[\![y]\!]$, we have $[\![uy]\!] = [\![vy]\!]$, and so by (7.6) and (7.12), $[\![\langle y\rangle(uy)]\!] = [\![\langle y\rangle(vy)]\!]$. But by (7.8–9), $[\![(\varepsilon u)]\!] = [\![\langle y\rangle(uy)]\!]$ and likewise with v in place of u, so $\varepsilon[\![u]\!] = \varepsilon[\![v]\!]$.

So $\langle D[X], \cdot, \varepsilon\rangle$ is a combinatory model. By the Lambda Model Theorem, $\langle D[X], \cdot, \varepsilon_2K, \varepsilon_3S\rangle$ is a lambda model. But

$\mathscr{C}[X] \models \varepsilon_2K = \varepsilon((B\varepsilon)\, K)$ by (6.1–2)

$= \langle x\rangle(B\varepsilon Kx)$ by (7.9)

$= \langle x\rangle(\varepsilon(Kx))$ by (6.1) and (7.12)

$= \langle x\rangle(\langle y\rangle(Kxy))$ by (7.9) and (7.12)

$= K$ by (7.1),

and a similar calculation using (7.2) shows that $\mathscr{C}[X] \models \varepsilon_3 S = S$, so $\mathscr{C}_\lambda[X]$ is this lambda model.

(ii) Equations (7.1–5) follow from the Combinatory Model Theorem (iv) since (7.1–5) are each of the form $u^{(\mathscr{C})} = v^{(\mathscr{C})}$ for convertible lambda terms u, v. ∎

ACKNOWLEDGMENTS

I thank Dana Scott for encouragement, technical suggestions, and explanation of his results, David McAllester whose excellent comments led to the present formulation of the Completeness Theorem, H. Barendregt and G. Longo for pointing out solutions to nearly all the questions I raised, Kim Bruce for patient close reading of several drafts, and Girard Berry, Dan Cooperstock, T. Fehlmann, Joe Halpern, Roger Hindley, Anne Mahoney, A. Obtulowicz, Rohit Parikh, Gordon Plotkin, David Park, Vaughan Pratt, Charles Rackoff, Rick Statman, Robert Street, Paul Weiss, A. Wiweger and E. Zachos for their comments and interest.

RECEIVED: July 19, 1982

REFERENCES

ACZEL, P. (1980), Frege structures and the notions of proposition, truth and set, *in* "The Kleene Symposium" (Barwise *et al.*, Ed.), pp. 31–60, Studies in Logic 101, North-Holland, New York.

ACZEL, P.(1981), Abstract models of the lambda calculus, *J. Symbolic Logic*, in press.

BARENDREGT, H. P. (1977), The type free lambda calculus, "Handbook of Mathematical Logic" (J. Barwise, Ed.), pp. 1091–1132, North-Holland, New York.

BARENDREGT, H. P. (1981), "The Lambda Calculus: Its Syntax And Semantics," Studies in Logic 103, North-Holland, New York.

BARENDREGT, H., AND KOYMANS, K. (1980), Comparing some classes of lambda-calculus models, *in* "To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism" (J. P. Seldin and J. R. Hindley, Eds., pp. 287–302, Academic Press, New York.

BARENDREGT, H. P., AND LONGO, G. (1980), Equality of λ-terms in the model T^ω, *in* "To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism," (J. P. Seldin and J. R. Hindley, Eds., pp. 303–338, Academic Press, New York.

BERRY, G. (1980), On the definition of lambda-calculus models, *INRIA Rapports de Recherche* **46**.

CHURCH, A. (1932/1933), A set of postulates for the foundation of logic, *Ann. of Math.* **33**, 346–366; **34**, 839–864.

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. (1930), Grundlagen der kombinatorischen Logik, *Amer. J. Math.* **52**, 509–536, 789–834.

ENGELER, E (1979), Algebras and combinators, Berichte des Insituts für Informatik 32, ETH, Zurich.

GORDON, M. J. C. (1979), "The Denotational Description of Programming Languages," Springer-Verlag, New York.

HYLAND, M. (1976), A syntactic characterization of the equality in some models for the lambda calculus, *J. London Math. Soc.* 361–370.

HINDLEY, R., LERCHER, B., AND SELDIN, J. (1972), "Introduction to Combinatory Logic," London Math. Soc. Lecture Note Series 7, Cambridge Univ. Press, London/New York.

KOYMANS, C. P. J.(1981), Models of the lambda calculus, Preprint 23, Department of Mathematics, University of Utrecht, Holland.

HINDLEY, R., AND LONGO, G. (1980), Lambda-calculus models and extensionality, *Z. Math. Logik Grundlag. Math.* **26**, 289–310.

LAMBEK, J. (1980), From λ-calculus to cartesian closed categories, *in* "To *H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*" (J. P. Seldin and J. R. Hindley, Eds.,) pp. 375–402, Academic Press, New York.

LANDIN, P. J. (1964), The mechanical evaluation of expressions, *Comput. J.* **6**, 308–20.

LANDIN, P. J. (1965), A correspondence between ALGOL 60 and Church's lambda-notation, Parts I and II, *Comm. ACM* **8**, 89–101, 158–65.

LONGO, G. (1981), Power set models of the lambda-calculus: Theories, expansions, isomorphisms, MIT/LCS/TM-207.

OBTULOWICZ, A.(1977), Functorial semantics of the type free λ-$\beta\eta$ calculus, "Fundamentals of Computation Theory," pp. 302–307, Lecture Notes in Computer Science 56, Springer-Verlag, New York.

OBTULOWICZ, 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, Sniadeckich 8, Skr. Poczt. 137, 00–950, Warszaw, Poland.

PLOTKIN, G. D. (1972), A set-theoretical definition of application, Memorandum MIP-R-95, School of Artificial Intelligence, University of Edinburgh.

PLOTKIN, G. D. (1974), The lambda calculus is ω-incomplete, *J. Symbolic Logic* **39**, 313–317.

PLOTKIN, G. D. (1975), Call-by-name, call-by-value, and the lambda calculus, *Theoret. Comput. Sci.* **1**, 125–159.

PLOTKIN, G. D. (1977), LCF Considered as a Programming Language, *Theoret. Comput. Sci.* **5**, 223–255.

PLOTKIN, G. D. (1978), T^{ω} as a universal domain, *J. Comput. System Sci.* **17**, 2, 209–236.

SCOTT, D. S. (1976), Data types as lattices, *SIAM J. Comput.* **5**, 522–587.

SCOTT, D. S.(1980a), Lambda calculus: Some models, some philosophy, *in* "The Kleene Symposium" (Barwise *et al.*, Ed.), pp. 381–421, Studies in Logic 101, North-Holland, New York.

SCOTT, D. S. (1980b), Relating theories of the λ-calculus, *in* "To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism" (J. P. Seldin and J. R. Hindley, Eds.), pp. 403–450, Academic Press, New York.

SCOTT, D. S. (1981), Lectures on a mathematical theory of computation, Oxford Univ. Computing Lab., Tech. Mono. PRG-19.

STOY, J. E. (1977), "Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory," MIT Press, Cambridge, Mass.

TURNER, D. A.(1979), Another algorithm for bracket abstraction, *J. Symbolic Logic* **44**, 267–270.

VOLKEN, H. (1978), "Formale Stetigkeit un Modelle des Lambda Kalkuls," P. D. thesis, ETH, Zurich.

WADSWORTH, C. (1976), The relation between computational and denotational properties for Scott's D_{∞}-models of the lambda-calculus, *SIAM J. Comput.* **5**, No. 3, 488–521.

Printed in Belgium