λ-LOGIC

by

Leonid Rudin

Technical Report #4521

May 1981

Computer Science

California Institute of Technology

Pasadena, California 91125

Submicron Systems Architecture Project

sponsored by

Defense Advanced Research Projects Agency
ARPA Order #3771

and monitored by

Office of Naval Research
Contract #N00014-79-C-0597

ACKNOWLEDGMENTS

It is my pleasure to thank my advisor, Professor James Kajiya, for being my teacher and colleague in this work. I also would like to thank him for making me aware that thorough studies of logical foundations are the only road for the theory of computation and programming languages in the long run.

I am grateful to Professor Alexander Kechris for our discussion on the language of λ-logic and for his helpful comments.

I wish to express my special gratitude to Denise Noël, who performed extremely difficult technical typing and never lost her patience and good humor.

ABSTRACT

The main aim of this paper is to formulate "natural" logical foundations for type-free λ-calculus. The importance of such foundations for analyzing arbitrary order computational properties of programs is emphasized.

λ-logic is a deductive system based on combinatory λ-terms. Its language is conceived by extending the set of λ-terms through the addition of new terms which are logical connectives. The model for λ-logic is Dana Scott's D_∞, which can be represented as a pseudo-Boolean algebra.

We present detailed proof that D_∞ can be constructed as a Heyting algebra, thus being a model for some Heyting intuitionistic logical system. Our result, briefly described above, poses new problems. In particular, the relation between algebraic models of computer languages and the algebraic model theory is of great interest if one wants to establish a logical framework for the verification of programs written in these languages.

INTRODUCTION

λ-logic is an axiomatic theory based on combinatory λ-terms. It differs from many existing axiomatic systems because of its type-free nature. The reader is assumed to have some understanding of the basic theory of λ-conversion.

Our Motivation and Historical Sketch follows this brief introduction. The reader who has had experience with λ-calculus and is familiar with the construction of the D_∞ model can skip most of Section 1. Section 2 will deal with problems arising in the logic which is based on combinators. We propose that D_∞ be used as a model for such a logic. The analogy of our approach with Von Neumann's resolution of set theory antinomies is discussed. Section 3 constitutes the proof of the hypothesis presented in Section 2 that D_∞ is a pseudo-Boolean algebra. So, Section 3 is the soul of this paper and is quite involved with lattice-theoretic proofs.

Finally, Section 4 describes the formalized theory of λ-logic in a rigorous manner. The proof of consistency of λ-logic concludes Section 4.

TABLE OF CONTENTS

MOTIVATION AND HISTORICAL SKETCH

The beginning of the mathematical science of computation as it is understood today goes back to John McCarthy's "Towards a mathematical science of computation" (McCarthy 1963). In that paper McCarthy sets up the framework and stated main problems of the field which is called now theoretical computer science. He asks, "What are the axioms and rules of inference of a mathematical science of computation?....Ideally we would like a mathematical theory in which every true statement about procedures would have a proof..." (1963:22) McCarthy also points out the necessity of a new logical theory in which neither the integers nor any other domain (e.g., strings of symbols) are given a special role. Of course, these were not the only problems pointed out by McCarthy, but without doubt they were fundamental. Almost twenty years have passed since the publication of this remarkable paper. Since then, many approaches to the science of computation have been developed.

Very important contributions were made by Zohar Manna and his students. In connection with the setting up of a logical framework, a paper by Manna and McCarthy (Manna, Z. and J. McCarthy 1970) is especially noteworthy. In this paper partial function logic was developed in the frame of 3-valued predicate calculus. Manna (Manna 1974) describes all kinds of inductive methods for proving properties of programs. He uses first-order propositional logic for proving theorems about flowchart programs. And for recursive programs, he uses the so-called "fixed point" approach. Such properties as termination, correctness, and equivalence are transformed into equivalent formulas of first-order predicate logic. Manna (Manna, Z. and A. Pnueli 1970:555) writes: "...we try to find an algorithm which will construct for a given program P (in the given language) a first-order formula W_p characterizing its execution. Then, for example, the problem of proving the convergence and correctness (with respect to a given assertion) of the program can be reduced to the problem of proving the validity of a formula constructed by the use of W_p." However, this method has its own irrefutable limitation. In the same paper (Manna, Z. and A. Pnueli 1970) Manna noticed that very important properties cannot be formalized by first-order formulas. He suggested the use of second-order calculus, applied by Cooper (Cooper 1969) for flowchart programs. The remaining question is, what if some programming concept has a higher than second-order nature?

In his 1976 ACM Turing Award lecture, Dana Scott calls the results described by Z. Manna (Manna 1974) as only the first chapter of the theory of computation. He writes, "The second chapter already includes procedures that take procedures as arguments -- higher type procedures -- and we are well beyond program schemes. True, fixed-point techniques can be applied....The semantic structure needed to make this definite is the function space." (1976:639) Here we come close to the so-called Scott-Strachey approach to programming language theory. In fact, our work is completely based on Scott's remarkable theory.

Starting in the early 60's, the λ-calculus has been used as an important tool "...in examining properties of programming languages precisely because it brings out very clearly the connection between a name and the entity it denotes, even in cases where the same name is used for more than one purpose." (Stoy 1977:xxiii -- Foreword by Dana S. Scott) λ-calculus models of programming languages were studied by

J. H. Morris in his Ph.D. dissertation (Morris 1968). Morris discovered the relationship between purely formal computation and the more informal notion of solutions for combinatorial equations. Morris (Morris 1968) investigated the paradoxical combinator Y and used it in constructing recursively defined functions. He showed that minimal fixed points produced by Y correspond exactly with the usually understood notion of recursive computation (1968:40-75) However, mathematical basis was needed to support this approach or, as a logician would say, a model was needed for consistency. The first to construct a model was Dana S. Scott. It was called D_∞ (D. Scott 1973). Historically it was the first solution to the isomorphic domain equation

$$D \simeq [D \rightarrow D].$$

In the conclusion of his paper, D. Scott writes: "Though the words 'calculus' and 'logic' do have general significance, I would propose calling the system of Church and Curry λ-algebra (or if you like: combinatory algebra)....What I have done is to introduce something new: limits and topology. Therefore, in analogy with classical mathematics, I would like to call the extended theory λ-calculus." (D. Scott 1973:186) In another paper D. Scott writes: "...the terminology 'combinatory logic' is still premature despite all the works of Curry and Fitch. We shall certainly establish connections with the usual kind of predicate logic, but it seems to this author that much remains to be done to determine whether these are the right connections or even especially useful ones." (D. Scott 1975:5) In line with Scott's ideas, Robin Milner (Milner 1972) designed and investigated his logic for computable functions (LCF). He demonstrates the soundness of the LCF with respect to models which are partially ordered domains (Milner 1973). However, the LCF approach is a version of the typed λ-calculus, which is too restrictive to describe high-level programming language concepts, especially when self-application is a desired property.

Joseph E. Stoy (1977) points out the importance of self-application for languages which allow commands themselves to be stored in locations. Besides, in a type-free system one does not worry about difficulties arising from type conflicts. So the unsolved question still remains: what are the connections between free λ-calculus and logic?

Because of the intimate relation between λ-calculus and computer science theory, we need but find the logic of λ-calculus to answer McCarthy's question: "What are the axioms and rules of inference of a mathematical science of computation?" (McCarthy 1963:22) Through analysis of the algebraic structure of the D_∞ model of λ-calculus, we attempt to solve precisely this last problem.

Interestingly enough, the solution which we propose in this work also partially answers another problem, listed as an open problem by H. F. Barendregt (1975). Barendregt asked: "What is a proper intuitionistic version of λ-calculus? And is there difficulty in showing constructively that models exist?" (Barendregt 1975:370) The answer to the last question leads us to a new development of the so called illative theory of combinatory logic.

We will formulate a logic based on combinatory λ-terms. Then we will prove its consitency by constructing an explicit model. To be more precise, we will look carefully at the construction of Scott's D_∞ model and prove that D_∞ has the right structure to be a model for our logic. In analogy with Dana Scott's naming of the extended theory of λ-algebra "λ-calculus," we will call this extended theory of λ-calculus "λ-logic."

1. TECHNICAL PRELIMINARIES

We assume the reader is familiar with the basic theory of λ-conversion and with the general nature of Scott's lattice-theoretic approach. Barendregt (1977) provides an excellent introduction to the subject. Also, he provides a full list of references. Nevertheless, we will give a quick review of the λ-calculus and D_∞ model. All necessary information on lattice theory is provided. However, for a more complete understanding of the relations between pseudo-Boolean algebras and formalized languages the reader can consult Birkhoff (1960) and Rasiowa and Sikorski (1970).

Concerning the characterization of the ordering relation of D_∞, which plays an important role in our work, Wadsworth's original paper on the subject (Wadsworth 1976) should be consulted. An introduction to illative combinatory logic is offered by Curry and Feys (1958), with alternative explanations of Russell's Paradox. For most of the review section 1, expositions from Barendregt (1977) and Wadsworth (1976) are used.

A. The λ-calculus.

1.1 Definition: Assuming denumerably many variables, x,y,z,...,x',y',z',..., we define a set of terms inductively as follows:

(i) Every variable is a term.

(ii) If A and B are terms, then AB is a term.

(iii) If A is a term and x is a variable, then λx.A is a term.
This operation is called an abstraction of A.

If parentheses are absent, then we assume application from the left to the right.

The scope of λx is determined similarly to the scope for quantifiers in the usual predicate logic. $\lambda x_1.(\lambda x_2(\ldots(\lambda x_n.A))\ldots)$ is abbreviated as $\lambda x_1\ldots x_n.A$. As usual, $\equiv$ means syntactical equivalence.

A variable occurs free in a term A if x is not in the scope of "$\lambda x.$"; otherwise, x is bound in A.

FV(A) is the set of all free variables in A. A is closed if $FV(A) = \phi$.

$[A/x]B$ results from the substitution of A for x in B, so that bound variables in B are changed when necessary to prevent capture of free variables of A: e.g., $[y/x]\lambda y.xy \equiv [y/x]\ \lambda z.xz \equiv \lambda z.yz$.

$R \equiv (\lambda x.A)B$ is called β-redex and

$R' \equiv [B/x]A$ is called β-contractum of R;

going from R to R' is called β-contraction;

going from R' to R is called β-abstraction.

A term is said to be in normal form if it does not contain a β-redex as a subterm.

1.2 As indicated by Wadsworth (1976), "=" is a substitutive equivalence relation. Equational calculus is given as follows: Let M and N be λ-terms.

Then (ρ) $M = M$

(δ) $M = N$; then $N = M$

(τ) $\dfrac{M = L,\ L = N}{M = N}$

(Subst.) $\dfrac{M = N}{C[M] = C[N]}$ for all contexts C[], where C[] is a term except for one missing subterm.

(α) $\lambda x.M = \lambda y.[y/x]M$, provided $y \notin FV(M)$

(β) $(\lambda x.M)N = [N/x]M$

(η) $\lambda x.Mx = M$ if $x \notin FV(M)$

We say $M = N$ iff it is provable from the axioms in (1.2).

1.2.1 Let $Y \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$. Then Y solves the problem: given any λ-term F, find a λ-term A such that $A = FA$: i.e., A is a fixed point of F.

Theorem

For any λ-term F there exists at least one fixed point A of F, namely

$$A = Y(F): \text{ i.e., } F(Y(F)) = Y(F).$$

This theorem is proved directly by applying Y to F and using the β-rule:

$$\begin{aligned} YF &= (\lambda f.(\lambda x.f(xx))(\lambda x.f(xx)))(F) \\ &= (\lambda x.F(xx))(\lambda x.F(xx)) \\ &= F((\lambda x.F(xx))(\lambda x.F(xx))) \\ &= F(YF)\text{: i.e., } YF \text{ is a fixed point.} \end{aligned}$$

Y is called a paradoxical combinator.

B. Construction of D_∞.

Because of the type-free nature of the application of λ-calculus, the domain of any interpretation must include a significant portion of its own function space. Thus semantics of this calculus is based on a solution of an isomorphism

$$(*) \quad D \simeq [D \rightarrow D],$$

where $[D \rightarrow D]$ denotes some suitable notion of function space from D to itself. Thus we need to construct a solution to the equation (*). Historically, a fist solution was found by D. Scott and was called D_∞. We now give some definitions and properties of D_∞. Again, for proofs the reader is referred to Barendregt (1977) and Wadsworth (1976).

1.3 Definition.

The partially ordered set $(D,\leq)$ is a complete lattice if $\forall X \subset D$, there exists a supremum $\sqcup X \in D$. Denote top $\top = \sqcup D$ and bottom $\bot = \sqcap D$ (i.e., largest and smallest elements of D). The supremum and infimum of $\{x,y\}$ are denoted as $x \sqcup y$, $x \sqcap y$.

1.4 Lemma

If D is a complete lattice, then each of its subsets X has an infimum: $\sqcap X = \sqcup\{z \mid z \leqq X\}$. Here $z \leq X$ iff $\forall x \in X$, $z \leq x$.

1.5 Definition.

A subset $X \subset D$ is directed iff $\forall x,y \in X \; \exists \; z \in X$, such that $x,y \leq z$.

1.6 Definition

A and B are complete lattices. A mapping f: $A \rightarrow B$ is called continuous iff $f(\underset{A}{\sqcup} X) = \underset{B}{\sqcup} f(X)$ for all directed $X \subset A$.

Here $f(X) = \{y = f(x) \mid x \in X\}$. $\underset{D}{\sqcup}$ means that a supremum is taken in set D; we usually omit D, since it will be clear from context.

Let $[A \rightarrow B]$ be the set of all continuous functions from A to B.

1.7 Definition.

Let D and D' be complete lattices. Let Φ: $D \rightarrow D'$, $\Psi : D' \rightarrow D$

(Φ,Ψ) is called a projection pair iff

(i) Φ,Ψ are continuous

(ii) $\forall x \in D$, $\Psi(\Phi(x)) = x$ or $\Psi \circ \Phi$ is an identity.

(iii) $\forall x \in D'$, $\Phi(\Psi(x)) \leq x$

1.8 Definition.

Let $(D,\sqsubseteq)$ be an arbitrary nontrivial complete lattice. Define $D_0 = D$, $D_{n+1} = [D_n \to D_n]$.

Define (Φ_n,Ψ_n) where $D_n \underset{\Psi_n}{\overset{\Phi_n}{\rightleftarrows}} D_{n+1}$, $n=0,1,2,\ldots$ where

$\Phi_0(x) = \lambda y \in D_0 . x, \quad x \in D_0$

$\Psi_0(x') = x'(\bot_{D_0}); \; x' \in D_1 ; \; \bot_{D_0}$ is $\sqcap D_0$.

And then, inductively, we define:

$\Phi_{n+1}(x) = \Phi_n \circ x \circ \Psi_n, \; x \in D_{n+1}$

$\Psi_{n+1}(x') = \Psi_n \circ x' \circ \Phi_n, \; x' \in D_{n+2}$

$$\begin{tikzcd}
\cdots & D_n \arrow[l] \arrow[d, "x"] & D_{n+1} \arrow[l, "\Psi_n"'] & D_{n+2} \arrow[l, "\Psi_{n+1}"'] & \cdots \arrow[l] \\
\cdots \arrow[r] & D_n \arrow[r, "\Phi_n"'] & D_{n+1} \arrow[r, "\Phi_{n+1}"'] \arrow[u, "x'"] & D_{n+2} \arrow[r] & \cdots
\end{tikzcd}$$

The inverse limit of the D_n's is called D_∞; i.e.,

$$D_\infty = \{\langle x_n \rangle_{n=0}^{\infty} \mid x_n = \Psi_n(x_{n+1}), \; x_n \in D_n\}.$$

1.8.1 Lemma.

Let D be a complete lattice. Then $[D \to D]$ is a complete lattice under pointwise ordering: i.e., $f \sqsubseteq g \Leftrightarrow \forall x \; f(x) \sqsubseteq g(x)$.

Corollary.

$\forall_n$, D_n is a complete lattice.

Definition.

Let x and $y \in D_\infty$; $x = \langle x_n \rangle_{n=0}^{\infty}$ and $y = \langle y_n \rangle_{n=0}^{\infty}$. Then we say $x \sqsubseteq y$ iff $\forall n \; x_n \sqsubseteq y_n$.

1.8.2 Theorem

D_∞ is a complete lattice under the componentwise partial ordering $\sqsubseteq$.

1.9 Lemma

(i) $\forall n \geq 0$, (Φ_n, Ψ_n) is a projection pair

(ii) $\forall n \geq 0$, Φ_n and Ψ_n are distributive.

1.10 Definition

Define a set of mappings inductively:

$$\{\Phi_{n\infty}: D_n \to D_\infty | n=0,1,2,\ldots\}$$

and

$$\{\Psi_{\infty n}: D_\infty \to D_n | n=0,1,2,\ldots\}$$

(i) Define $\Phi_{0\infty}: D_0 \to D_\infty$

$\forall x_0 \in D_0 \quad \Phi_{0\infty}(x_0) = \langle y_n \rangle_{n=0}^{\infty} \in D_\infty$

such that $\quad y_0 = x_0,\ y_{n+1} = \Phi_n(y_n) \quad n=0,1,2,\ldots$

i.e., $\quad \Phi_{0\infty}(x_0) = (x_0, \Phi_0(x_0),\ \Phi_1 \circ \Phi_0(x_0), \ldots).$

And define $\Psi_{\infty 0}: D_\infty \to D_0$ as follows:

$\forall x = \langle x_n \rangle_{n=0}^{\infty} \in D_\infty,\ \Psi_{\infty 0}(x) = x_0.$

(ii) Let $\Phi_{n+1\infty}: D_{n+1} \to D_\infty$.

Then define $\Phi_{n+1\infty}(x) = \Phi_{n\infty} \circ x \circ \Psi_{\infty n},\ x \in D_{n+1}$

and $\Psi_{\infty n+1}: D_\infty \to D_{n+1}$ as

$\Psi_{\infty n+1}(x') = \Psi_{\infty n} \circ (x')^{\star} \circ \Phi_{n\infty},\ x' \in D_{\infty}$

In the last equation $(x')^{\star}$ is used as a function on D_{∞}. It is defined in (1.11).

1.11 Definition

Let $x \in D_{\infty}$, $y \in D_{\infty}$: i.e., $x = \langle x_n \rangle_{n=0}^{\infty}$ and $y = \langle y_n \rangle_{n=0}^{\infty}$. Then $x^{\star}$ is a function from $D_{\infty} \to D_{\infty}$, such that for any y, $x^{\star}(y) = z \in D_{\infty}$, where $z = \langle z_n \rangle_{n=0}^{\infty}$ and

(i) $z_0 = \bigcup\{x_1(y_0), \Psi_0(x_2(y_1)), \ldots, \Psi_0 \circ \Psi_1 \circ \ldots \Psi_n(x_{n+2}(y_{n+1})) \ldots\}$

(ii) $z_n = \bigcup\{x_{n+1}(y_n), \Psi_n(x_{n+2}(y_{n+1})), \ldots \Psi_0 \circ \Psi_1 \circ \ldots \circ \Psi_{n+k}(x_{n+k+2}(y_{n+k+1})), \ldots\}$.

1.12 Lemma

Let $x \in D_{\infty}$. Then $x^{\star} \in [D_{\infty} \to D_{\infty}]$; i.e., $x^{\star}$ is a continuous function from D_{∞} to D_{∞}.

1.13 Lemma

$\forall f \in [D_{\infty} \to D_{\infty}]\ \exists\, x \in D_{\infty}$, such that $f = x^{\star}$.

So we see that (1.11) and (1.12) provide the following embedding: $x \to x^{\star}$ is the mapping $D_{\infty} \to [D_{\infty} \to D_{\infty}]$. We call it Φ. And Lemma (1.13) gives the mapping from $[D_{\infty} \to D_{\infty}]$ to D_{∞}. Call it Ψ.

1.14 Lemma

D_{∞} is homeomorphic to $[D_{\infty} \to D_{\infty}]$ under a pair of isomorphisms (Φ, Ψ), i.e.,
$$\begin{tikzcd} D_{\infty} \arrow[r, shift left, "\Phi"] & {[D_{\infty} \to D_{\infty}]} \arrow[l, shift left, "\Psi"] \end{tikzcd}.$$

1.15 Definition

$$D_n^\infty = \{\Phi_{n\infty}(x) \mid x \in D_n\} \sqsubseteq D_\infty.$$

D_n^∞ is a subspace of D_∞, and obviously $D_n^\infty \simeq D_n$. Let $P_n \equiv \Phi_{n\infty} \circ \Psi_{\infty n} : D_\infty \to D_n^\infty$: i.e., a projection from the space D_∞ into the subspace D_n^∞. For convenience, we write x_n for $P_n(x)$, $x \in D_\infty$, and we write $x(y)$ for $x^\star(y)$ and f for $\Psi(f)$.

1.16 Lemma

For all x and y $\in D_\infty$:

(i) $\bot(y) = \bot = \sqcap D_\infty$

(ii) $\top(y) = \top = \sqcup D_\infty$

(iii) $x = y$ iff $x_n = y_n$ for all $n \geq 0$.

1.17 Lemma

$$D_0^\infty \sqsubseteq D_1^\infty \sqsubseteq D_2^\infty \sqsubseteq \ldots \sqsubseteq D_n^\infty \sqsubseteq \ldots \sqsubseteq D_\infty.$$

C. D_∞ as a model for λ-calculus

Since $D_\infty \simeq [D_\infty \to D_\infty]$, it is an appropriate model for a language where application is allowed without type limitations. Of course, then D_∞ is a possible model for λ-calculus, since if A and B are any terms, then A(B) and B(A) are perfectly defined terms. Thus, we should be able to map our language into the D_∞ model.

1.18 Definition

(i) Let ρ: {variables} $\to D_\infty$, ρ is called a valuation in D_∞.

(ii) For $d \in D_\infty$ and x a variable, $\rho(d/x) = \rho'$, where ρ' is a valuation in D_∞, such that $\rho'(y) = \rho(y)$ if $y \not\equiv x$ and $\rho'(x) = d$.

(iii) The interpretation ℓ of any term A in D_∞ under ρ is denoted $\ell[\![A]\!](\rho)$ and is a mapping from the set of λ-terms into the D_∞. It is defined inductively:

(S1) $\ell[\![x]\!] (\rho) = \rho(x)$, where x is a variable.

(S2) $\ell[\![MN]\!] (\rho) = \Phi(\ell[\![M]\!] (\rho)) (\ell[\![N]\!] (\rho))$.

(S3) $\ell[\![\lambda x.M]\!] (\rho) = \Psi(\lambda d \in D_\infty . \ell[\![M]\!] (\rho(d/x)))$.

1.19 Theorem

Axioms (ρ), (δ), (τ), (subst.), (α), (β), (η) of (1.2) are valid formulas under the D_∞ interpretation.

1.20 Theorem

D_∞ is a model for λ-calculus.

D. A characterization of $\sqsubseteq$ in D_∞.

The following is due to Wadsworth (1976).

1.21 Definition

Each λ-term M is of the form $M \equiv \lambda x_1 \ldots x_n . (\lambda x.P) QM_1 \ldots M_m$ or $M \equiv \lambda x_1 \ldots x_n . x_i A_1 A_2 \ldots A_k$. ($\lambda$x.P) is called the head redex of M. In the second case x_i is the head variable of M and M is said to be in a head normal form (compare with the definition of the normal form in (1.1)).

1.22 Definition

Let M and N be terms. Then we say $M \sqsubseteq N$ iff $\ell[\![M]\!](\rho) \sqsubseteq \ell[\![N]\!](\rho)$ for any valuation $\rho \in (Var \rightarrow D_\infty)$. Var is a set of all variables.

Lemma

$$M \sqsubseteq N \iff C[M] \sqsubseteq C[N] \text{ for all contexts } C[\ ].$$

1.2 3 Definition

We say that $M \leq_W N$ iff, for all contexts C[ ]:

if C[M] has a head normal form,
then C[N] has the same head normal form.

Here $\leq_W$ stands for Wadsworth's ordering relation.

1.2 4 Theorem (Wadsworth)

For all terms M and N

$$M \leq_W N \iff M \sqsubseteq N.$$

Here we would like to emphasize the importance of this theorem. Theorem (1.2 4) establishes the relation between the purely syntactical concept of the "head normal form" and the semantical relation "$\sqsubseteq$" in D_∞. Thus, we have this important link between syntactical form and semantical content. Historically, this kind of property made mathematics a deductive science and created the science of mathematical logic. With this in mind, we attempt to explore this link and the deductive reasoning system behind it.

2. D_∞ VERSUS RUSSELL'S PARADOX

A. Relation to Von Neumann's set theory.

In June 1901 Bertrand Russell discovered a paradox which shook the mathematical world. In his letter to the great logician Gottlob Frege he communicates: "Let W be the predicate: to be a predicate that cannot be predicated of itself. Can W be predicated of itself? From each answer its opposite follows." (Russell, "Letter to Frege," 1967:125)

This may be written as follows: Let F(f) be the property of properties f defined by the equation

2.1 $F(f) = \sim f(f)$, where "~" is the symbol for negation.

Then, substituting F for f in (2.1), we get

2.2 $F(F) = \sim F(F)$

i.e., F has property F iff F does not have property F, which obviously is a contradiction under the assumption that F(F) is a proposition (true or false).

The usual explanation of this paradox is that such a definition of F (2.1) or, at any rate, F(F), is "meaningless." In the case of set theory, this paradox is formed as follows:

2.3 Let $R = \{X: X \notin X\}$.

Now, we ask, "Does R itself satisfy the condition $X \notin X$?" If $R \notin R$, then by definition (2.3) $R \in R$; then, again by (2.3) $R \notin R$. So both assumptions lead us to a contradictory conclusion.

The above argument is called Russell's Paradox.

Thus, suddenly mathematicians were faced with the necessity of revising Cantor's naive ideas of set theory. The first solution to this problem was presented by B. Russell in his theory of types (1967:150-82). He introduced typed propositional functions and typed arguments. Thus in (2.1) the definition of F would not make any sense because f(f) is not a proposition since f can be applied only to arguments of type lower than itself. The types are mutually exclusive, and thus no reflexive fallacies are possible.

However, one might think about function I, which is an identity for any argument x

2.4 $I(x) = x$, thus $I(I) = I$.

It makes sense and does not constitute a paradox. Nonetheless, it is a forbidden expression in Russell's theory. Moreover, many functions in mathematics are defined through self-referencial definitions.

Zermelo's axiomatization of set theory (Zermelo 1967) appeared shortly after Russell's theory of types. Unfortunately, since Zermelo pays no attention at all to the underlying logic, the notion of "proposition" is left unspecified. Thus the formulation of Russell's Paradox in (2.1) is not adequately resolved in Zermelo's system, while that in (2.3) is.

What we need is a reasonably flexible definition of the notion of a <u>function</u>. By prohibiting f(f) we exclude paradoxes -- "But there is evidently something about the preceding intuitive argument (2.1) which is not explained by such exclusions." (Curry and Feys 1958: 4)

Looking at (2.1) [$F(f) = \sim f(f)$] more carefully, we notice that not only is a self-applied $f(f)$ construction involved, but also "~", a logical sign. This suggests that something could be wrong with our naive use of the operation ~ of negation: i.e., the underlying logic should be rigorously defined. Finding an appropriate underlying logic might provide us with an alternative explanation of Russell's Paradox and permit a more flexible definition of function -- e.g., one that allows self-application for a great variety of functions.

Form (2.1) of Russell's Paradox can be easily expressed in our λ-notation. Let N represent a negation sign: i.e., we assume that the negation is a normal combinator and all λ-calculus conversion rules can be applied to it. Then (2.1) is equivalent to defining

2.5 $F = \lambda f.N(ff)$.

That is, we permit the λ-abstraction of $N(ff)$.

2.6 Then $F(F) = (\lambda f.N(ff))(F) = N(F(F))$.

That is, we arrived at a contradiction. So, the essential assumption here is whether the λ-abstraction in (2.5) is consistent with our system. In fact, by theorem (1.2.1.), any λ-term has a fixed point derived by application of the paradoxical combinator Y:

(2.7) $N(YN) = YN$.

Let $YN = A$. Then $N(A) = A$.

That is, the system of our logic is inconsistent. (This is why Y is called paradoxical).

Now, return to the third possible resolution of the paradox in form (2.3), formulated for sets -- namely, the set theory proposed by John Von Neumann in 1925. We are refering here to the original approach of John Von Neumann, and not to the more well known theory of sets and classes (NBG: a somewhat changed version of Neumann's set theory by P. Bernays and K. Gödel). The main difference between both theories is that Neumann used "function" as the basic notion, rather than using set and class.

Von Neumann's theory is then much closer to the ideas of combinatory logic than the NBG or Zermelo systems. The notion of "set," can be easily included in the notion of "function" (a set can be identified with its characteristic function).

To build a consistent model of set theory Neumann considered two domains of objects: the domain of arguments and the domain of functions. The two domains are not identical, but they can overlap. In other words, there exist "argument-functions" which belong to both domains. The main question is

2.8 "What 'functions' are at the same time 'arguments'?" (Von Neumann, in van Heijenoort, ed., 1967:397)

All of them? If so, we easily derive Russell's and many other antinomies of naive set theory. Neumann then imposes certain restrictions on his domains and gives a condition under which "function" will fail to be an argument. He speaks of "I-objects," "II objects," and "I-II objects" instead of "arguments," "functions," and "argument-functions," respectively. Of course, "I-II objects" are the hardest to define. It could be required that every I-object be an I-II object: i.e., each object is also a function.

We shall not present here the details of the model of set theory proposed by Von Neumann (in van Heijenoort, ed., 1967). The main point has been to indicate the cornerstone of that theory -- namely, question (2.8), the resolution of which helps to avoid antinomies and gives a consistent theory.

Return now to the formulation of Russell's Paradox in (2.7). If we consider λ-terms to be functions (as Church originally intended), a strong analogy with Von Neumann's set theory is apparent. Then, maybe, the resolution of (2.8), reformulated for λ-terms (functions), will explain what is wrong in (2.7).

Our view is that YN must be excluded from consideration as a λ-term because it is simply "meaningless." But then what is the definition of "meaning" for a λ-term? Looking ahead, we note that in Von Neumann's terminology N simply is not a "function-argument," but just a "II-object." However, in Von Neumann's theory the class "I-II" is precisely defined. Unfortunately, this definition does not work in our case. But then the question is: can we rigorously define a class of λ-terms which is analogous to Von Neumann's class of "function-arguments"? The answer is yes!

The subject of our discussion is a kind of *logic* based on λ-terms as propositions. In fact, this idea is not new. The so called *illative combinatory logic* started its development as a system of formal logic in Church's works. This system used λ-calculus as its algebraic language and, most important, it contained *implication*, *negation*, etc. However, this sytem was proved to be inconsistent (e.g., (2.7)). Its inconsistencies clearly demonstrated that illative combinatory logic was not as simple as originally supposed. The name "illative" was used because such concepts as quantification, implication, etc., were present in the system, bringing it closer to logic in the classical sense. The word "illation" comes from the Latin "inferre," i.e., a logical deduction. So, instead of constructing a completely new notation, we start as in the usual exposition of ICL (illative combinatory logic), e.g., as done by H. B. Curry and R. Feys (1958) or R. Hindley, B. Lercher, and J. Seldin (1972). But this parallel does not go far. Immediately after exposing that the system fails to work, we propose a different approach from not only the so called theory of functionality but also from other systems as well. We don't accept the theory of functionality because it is actually the same as Milner's LCF (Milner 1972). The shortcomings of this system were explained in our Motivation section.

Other systems, in our view, are simply inapplicable to the mathematical science of computation, although they have their own significance as an ultimate foundation of logic. As far as the mathematical science of computation is concerned, we feel that our approach is "natural."

B. Inconsistency of illative logic (Hindley, Lercher and Seldin 1972:102)

In addition to λ-calculus equational theory (1.2), we want to consider statements of the form

2.9 $\vdash X$, where X is a λ-term.

(2.9) is called an assertion and is considered to be a statement. The intuitive meaning of "assertion" is the same as "formula" in usual logic. We postulate the following rule:

2.10 Rule Eq. $X = Y$ & $\vdash X$, then $\vdash Y$.

2.11 Definition.

We abbreviate

$\vdash A_1$ &...& $\vdash A_n \Longrightarrow \vdash X$ as $A_1, A_2, \ldots, A_n \vdash X$.

Now we adjoin new terms whcih represent usual logical connectives and quantifiers to the set of λ-terms. They are P, N, and some others indicated later.

Example: PXY in our metalanguage means $X \Longrightarrow Y$ (X implies Y), and NX means ~X (negation of X). We would like to postulate rules which would be similar to those used when implication and negation are used in the orthodox sense. At least we would like to have the following:

2.12 Modus ponens. $PXY, X \vdash Y$ (i.e., $X \Longrightarrow Y, X \vdash Y$)

and

2.13 $\vdash P(PX(PXY))(PXY)$ (i.e., $\vdash (X \Longrightarrow (X \Longrightarrow Y)) \Longrightarrow (X \Longrightarrow Y)$)

Since combinatorial notation is not customary to our eye, we abbreviate

2.14 (i) $X \Longrightarrow Y \equiv PXY$

(ii) $\sim X \equiv NX$

2.14.1 Definition

We will refer to this sytem of equational calculus -- (1.2) together with (2.9), (2.10), (2.11), (2.12), (2.13) -- as the B-system.

Now, it is easy to show that this sytem is already inconsistent in the sense that $\vdash X$ is provable for every λ-term X! Let X be any λ-term.

2.15 Let $Z = \lambda z.(zz \Rightarrow (zz \Rightarrow X))$ $z \notin FV(X)$.

2.16 Let $M = ZZ$.

From (2.15) and β-reduction and (2.16)

2.17 $M = ZZ = (\lambda z.(zz \Rightarrow (zz \Rightarrow X)))\ Z$

$= ZZ \Rightarrow (ZZ \Rightarrow X) = M \Rightarrow (M \Rightarrow X)$.

Replacing X by M and Y by X in (2.13) we get

2.18 $\vdash (M \Rightarrow (M \Rightarrow X)) \Rightarrow (M \Rightarrow X)$.

But $(M \Rightarrow (M \Rightarrow X)) = M$ by 2.17. Then, using Eq.,

2.19 $\vdash M \Rightarrow (M \Rightarrow X)$.

From (2.19) and (2.18) and modus ponens (2.12),

2.20 $\vdash M \Rightarrow X$.

Then, by (2.19) and (2.17),

2.21 $\vdash M$.

Hence, from (2.20), (2.21) and modus ponens,

2.22 $\vdash X$.

Since X is an arbitrary term, the B-system is <u>inconsistent</u>. What is wrong?

C. D_∞ *as a model of illative logic.*

The close relationship between logic and algebra is a well established topic in modern mathematics. Boole discovered that logical notions such as "proposition," "not," "or," "and" can be interpreted in some special class of algebras, which we now call Boolean. It has been realized that after the identification of equivalent formulas in a formalized theory (e.g., first order predicate calculus), the set of all formulas can be embedded into a lattice. This lattice is called the model of this formalized theory. The kind of lattice used for embedding depends on the type of logic under consideration, and vice versa. Such an embedding is called an interpretation.

For classical logic Boolean lattices are used. For intuitionistic logic, as formulated by Heyting, the corresponding model was proven to be a pseudo-Boolean lattice with zero, or a Heyting algebra. The algebra of subsets of topological spaces happened to be a suitable model for modal propositional calculus. H. Rasiowa and R. Sikorski (1970) present in uniform manner the algebraic approach to classical, intuitionistic, modal and positive logic. This approach was used significantly by computer scientists for the creation of the denotational semantics theory. A good example is section 1.C., where D_∞ is proven to be an algebraic model for λ-calculus. There, λ-terms were interpreted as points in D_∞.

Now, if we extend the set of λ-terms by adding new terms such as N-negation and P-implication, it is in the spirit of an algebraic approach to interpret these new terms (which are logical connectives in our metalanguage) somehow in our D_∞ model.

Thus, extension for a mapping $\ell[[\ ]]$ (1.18) is needed. Assume that we extended interpretation to N and P. Since the structure of D_∞ is a much clearer matter than the hidden structure of the deductive B-system, it can be used to explain inconsistency (2.22), and, of course, any other paradox which involves this system (e.g., Russell's Paradox (2.7)). By theorem (1.8) D_∞ is a complete lattice under componentwise partial ordering. In the algebraic semantics of classical logic (CL) negation, "~," is usually interpreted as a complement operation in a corresponding Boolean model B. It also has an operation corresponding to implication. The interpretation of the implication is given by:

2.23 $[[\Rightarrow]] = \lambda xy\ .\ x' \cup y$, where $x, y \in B$ and "'" is a complement operation. That is $[[\Rightarrow]] : B^2 \rightarrow B$.

For intuitionistic logic (IL), the model is Heyting algebra H. The negation "~" and consequence "=>" operations in IL differ from CL's implication according to IL's and CL's systems of axioms. Therefore these operations assume a different interpretation than in Boolean algebra.

The notion of pseudo-complementation is needed to define negation and implication. We will define it later, and now assume it exists and it is the mapping $\underset{H}{\Rightarrow}: H^2 \rightarrow H$. If 0 is the zero element of H, then

2.24 we define $\underset{H}{\sim}: H \rightarrow H$ by $\underset{H}{\sim} x \equiv (x \underset{H}{\Rightarrow} 0)$.

It turns out that such an interpretation of logical connectives translates axioms of IL into valid statements about algebra H. This means that the metatheory of IL coincides with the theory of pseudo-Boolean algebras. In particular, the existence of this model for IL means the consistency of IL as a logical system.

The meta meanings for N and P in 2.B are negation and implication. Then (2.12) becomes a modus-ponens rule and (2.13) translates into one of the axioms of IL.

In (2.B) we abbreviate PXY as $X \Rightarrow Y$ and NX as $\sim X$. Here X and Y are not propositions of IL or CL, but arbitrary λ-terms. Is such an abbreviation just a notation to which our eye is accustomed, or is there some link between intuitionistic logic and logic based on λ-terms? Metanotation for the deduction rule (2.12) and axiom (2.13) suggests that the second assumption might be true. Can we define this link in precise mathematical terms? Yes! But we need some preliminary assumption, which we will prove in the next section.

2.25 Hypothesis: D_∞ is a Heyting albegra. Now, operation of the relative pseudo-complementation is well defined in D_∞ and thus it can be used for interpreting negation and implication, in a manner similar to that carried out by H. Rasiowa and R. Sikorski (1970) for the propositional IL. All needed definitions concerning Heyting algebra appear at the beginning of the next section.

Hypothesis (2.25) provides us with the natural (we claim) interpretation of the λ-calculus extended by N and P. Let "$\Rightarrow$" be pseudo-complementation in D_∞. Let $\perp$ be the smallest element in D_∞. Now we can extend the structural induction list (1.18) of S1, S2, S3 interpretation rules by following S4 and S5:

Let X, Y be already properly interpreted by $\ell[[\ X\ ]](\rho)$ and $\ell[[\ Y\ ]](\rho)$. Then

2.27 ((S4) $\ell[[\ PXY\ ]](\rho) = (\ell[[\ X\ ]](\rho) \Rightarrow \ell[[\ Y\ ]](\rho)) \in D_\infty$

2.28 (S5) $\ell[[\ NX\ ]](\rho) = (\ell[[\ X\ ]](\rho) \Rightarrow \perp) \in D_\infty$

In (S4) the interpretation of PXY is the pseudo-complement of $\ell[[\ X\ ]](\rho)$ relative to $\ell[[\ Y\ ]](\rho)$, which exists by our hypothesis. In (S5) the value of N(X) is interpreted as a pseudo-complement of the $\ell[[\ X\ ]](\rho)$ in D_∞. Again, it is well defined due to hypothesis (2.25).

It thus appears that at last we have natural restriction on the notion of "λ-term" or, corresponding to it, the notion of "proposition" for the illative B-system. The question, what is a proposition and what is not, is a decisive factor in the settlement of Russell's Paradox.

We postulate the following Principle.

2.29 Principle

A term X is meaningful or is a proposition if and only if for any ρ it has the interpretation $\ell[[\ X\ ]](\rho) \in D_\infty$.

Now we will use the Principle (2.29) to show that the reasoning in our demonstrations (2.7) and (2.15 - 2.22) was erroneous from the point of view of the semantics of D_∞. By (S2) in (1.18) $\ell[[\ AB\ ]](\rho) = \Phi(\ell[[\ A\ ]](\rho))(\ell[[\ B\ ]](\rho))$. Let Y be the paradoxical combinator defined in (1.2.1) and N be the negation introduced by (S5). Then

2.30 $\ell[[\ YN\ ]](\rho) = \Phi(\ell[[\ Y\ ]](\rho))(\ell[[\ N\ ]](\rho))$.

Well, $\Phi(\ell[[\, Y \,]](\rho))$ is well defined and is known as Tarski's fixed point operator. However, $\ell[[\, N \,]](\rho)$ was never defined in our system. Thus for (2.30) to make sense, at least N should be interpreted. But can N be interpreted without violating the Principle (2.29)? Let us try.

Assume that N is a meaningful term. Then all axioms of λ-calculus can be applied to it. Therefore, by the η-axiom from (1.2) we can derive

3.31 $(\lambda x \,.\, Nx) = N$ (by the extensionality property), whence

2.32 $\ell[[\, N \,]](\rho) = \ell[[\, \lambda x \,.\, Nx \,]](\rho)$.

Then, by (S3) in (1.8) it follows that

2.33 $\ell[[\, N \,]](\rho) = \Psi(\lambda d \in D_\infty \,.\, \ell[[\, Nx \,]](\rho(d/x)))$ which, by (S5) of (2.28)

$= \Psi(\lambda d \in D_\infty \,.\, d \Rightarrow \bot)$.

This last expression is well defined if and only if function $(\lambda d \in D_\infty \,.\, d \Rightarrow \bot)$ is an element of $[D_\infty \to D_\infty]$: i.e., it should belong to the domain of the mapping Ψ. This is, we claim, the hidden source of Russell's Paradox.

2.34 Let $N_\infty = \lambda d \in D_\infty \,.\, d \Rightarrow \bot$.

It follows from the definition of continuous function (1.6) that if f is continuous, then it is monotonic or order preserving.

$\bot$ and $\top$ denote, respectivelly, minimal and maximal elements of D_∞. So $\bot < \top$.

2.35 But $N_\infty(\bot) > N_\infty(\top)$ (i.e., not order preserving) because

$N_\infty(\bot) = \bot \Rightarrow \bot = \top$ and

$N_\infty(\top) = \bot \Rightarrow \top = \bot$ (see definition of "$\Rightarrow$" in the next section).

Thus N_∞ is not an order preserving function and therefore <u>is not continuous</u>. Whence, $N_\infty \notin [D_\infty \to D_\infty]$ and $\Psi(N_\infty)$ in (2.33) is not well defined.

Thus $\ell[\![N]\!](\rho)$ does not have a meaningful interpretation in D_∞; hence, $\ell[\![YN]\!](\rho)$ in (2.30) is not meaningful. Therefore, according to our <u>Principle</u> (2.29) Y(N) is <u>meaningless</u>. It is not a proposition. Thus the paradox disappears and the argument becomes nothing more than a proof that $\ell[\![N]\!](\rho)$ is not continuous and therefore N cannot be used as an argument for another term without violation of the denotational rules S1 – S5. However, N can be used as a function since $\ell[\![N]\!](\rho)$: $D_\infty \to D_\infty$: That is, it is a function and, being applied to any meaningful (by Principle) term, it produces a point in the D_∞: in other words, another meaningful term.

The similarity with J. Von Neumann's solution exposed earlier in our work is striking. Generally speaking, our Principle is an answer to Von Neumann's question (2.8) of "What functions are at the same time 'arguments.'" Our answer is: continuous functions on D_∞ are at the same time arguments.

Let us show how our Principle resolves paradox (2.5) - (2.6) and paradox (2.15)-(2.22) of R. Hindley, B. Lercher and J. Seldin (1972).

In (2.5) we defined $F = \lambda f \, . \, N(ff)$. Then $\ell[\![\lambda f \, . \, N(ff)]\!](\rho)$ $= \lambda d \in D_\infty \, . \, d(d) \Rightarrow \bot$.

2.37 Let $F_\infty \equiv (\lambda d \in D_\infty \, . \, d(d) \Rightarrow \bot) \in (D_\infty \to D_\infty)$. Then $F_\infty(F_\infty)$ does not have denotation since F_∞ is not continuous. The fact that F_∞ is not continous and therefore cannot be used as an argument can be easily seen by applying F_∞ to $\bot$ and $\top$, as done in (2.34) and (2.35) for N_∞.

2.38 $F_\infty(\bot) = (\bot(\bot) \Rightarrow \bot) = (\bot \Rightarrow \bot) = \top$.

2.39 $F_\infty(\top) = (\top(\top) \Rightarrow \bot) = (\top \Rightarrow \bot) = \bot$.

That is $F_\infty(\bot) > F_\infty(\top)$. Hence, F is not a continuous mapping and therefore is meaningless by the Principle. So, the abstraction of N(ff) in (2.5) which rendered the term $\lambda f.N(ff)$ was an illegal step leading to antinomy (2.6). Therefore Russell's Paradox in the form (2.5) - (2.6) is not a paradox any more: q.e.d.

2.40 The explanation for the paradox (2.15) - (2.22) is based on the same sort of arguing: at some point in (2.15) - (2.22) a meaningless term was introduced and later used for deriving a contradiction. Namely, in (2.15), $Z = \lambda z.(zz \Rightarrow (zz \Rightarrow X))$ is a meaningless term according to the Principle. It is easy to see (by S5) that if Z had a well defined denotation $\ell[[\ Z\ ]](\rho)$, the denotation whould be

2.41 $Z_\infty \equiv \ell[[\ Z\ ]](\rho) = \lambda d \in D_\infty.(d(d) \Rightarrow (d(d) \Rightarrow \ell[[\ X\ ]](\rho)))$

2.42 By the definition $\bot < \top$.

2.43 However, by (1.16) $Z_\infty(\bot) = (\bot(\bot) \Rightarrow (\bot(\bot) \Rightarrow \ell[[\ X\ ]](\rho)))$

$= \bot \Rightarrow (\bot \Rightarrow \ell[[\ X\ ]](\rho)) = \top$.

2.44 Now $Z_\infty(\top) = (\top(\top) \Rightarrow (\top(\top) \Rightarrow \ell[[\ X\ ]](\rho)))$, which, by (1.16)

$= \top \Rightarrow (\top \Rightarrow \ell[[\ X\ ]](\rho))$,

which, by the definition of "$\Rightarrow$"

$= \top \Rightarrow \ell[[\ X\ ]](\rho) = \ell[[\ X\ ]](\rho)$

2.45 We conclude that $F_\infty(\bot) > F_\infty(\top)$; unless $\ell[[\ X\ ]](\rho) = \top$ for any ρ: Var $\rightarrow D_\infty$. Thus this function is not continuous.

It follows that Z in (2.15) is meaningless according to the Principle. Thus the arguing in (2.15) - (2.22) was based on a false presupposition, q.e.d.

Note: If $\ell[[X]](\rho) = \top$ for any (ρ), then the conclusion $\vdash X$ in (2.22) is not a contradiction as it is claimed, and, on the contrary, is a "true statement," according to the definition of "truth" which will be introduced in subsequent sections.

In order to avoid contradictory arguings of the type we exposed in (2.5), (2.7), (2.15) - (2.22), and in order to make our logical B-system consistent, we have to reformulate the definition of the λ-term. This means that we are going to change the inductive definition in 1.1.

We proceed with the following definition.

2.46 Assume denumerably many variables x,y,z,... and define the set of <u>formulas</u> inductively as follows:

1. Each variable is a formula.
2. If M and N are formulas, so are A(B), PAB, NA, NB.
3. If x is a variable and M is a formula, then the abstraction $(\lambda x.M)$ satisfies the Principle (2.29).
4. An expression is a formula if and only if it was constructed by application of (1), (2), (3).

The conventions on parentheses are the same as in (1.1).

2.47 Equational calculus (1.2) remains the same except for the assumptions on M and N. Here we assume that M and N are formulas. Now, let us move to the next section, where we prove hypothesis (2.25).

3. D_∞ AS A PSEUDO-BOOLEAN ALGEBRA.

All proven results in this section are due to the author. The sources of the theory of pseudo-Boolean algebras are Garrett Birkhoff (1960) and H. Rasiowa and R. Sikorski (1970) Knowledge of section 1.B is essential for understanding the theory developed henceforth.

3.1 Definition.

Let D be a lattice and a,b $\in$ D; then $a \cup b$ and $a \cap b$ are the supremum and infimum, respectivelly, of {a,b}.

3.2 Definition.

Let a,b be elements of a lattice $(D,\leq)$. An element c $\in$ D is said to be the pseudo-complement of a, relative to b, if c is the greatest element, such that $a \cap c \leq b$. The pseudo-complement of a, relative to b, is denoted, if it exists, by the symbol

$$a \Rightarrow b$$

3.3 Lemma.

For every $x \in D$, $x \leq a \Rightarrow b$ iff $a \cap x \leq b$.

3.4 Definition.

A lattice D is said to be relatively pseudo-complemented (r.p.c.) if $a \Rightarrow b$ exists for all elements a,b $\in$ D.

3.5 Definition

Let D be a pseudo-complemented lattice with the zero element (minimal element of the D), and let a $\in$ D. Then element $(a \Rightarrow \perp)$ is called the pseudo-complement of a, and denoted a^*. It follows from Definition (3.2) that a^* is the greatest element disjoint from a.

Also, it is easy to see that for any $a \in D$, $a \Rightarrow a$ is the unit element $\top$ (greatest element in D).

3.6 Definition.

A lattice D is said to be distributive if for all $a,b,c \in D$,

3.7 $a \cap (b \cup c) = (a \cap b) \cup (a \cap c)$, and

$a \cup (b \cap c) = (a \cup b) \cap (a \cup c)$.

3.8 Lemma.

If lattice D satisfies at least one of the identities (3.7), then D is distributive. Distributivity can be generalized for an infinite case.

3.9 Definition.

Lattice D is infinitely distributive iff for all $x, y \in D$ and $\{y_\beta | \beta \in B\} \subset D$ and $\{x_\alpha | \alpha \in A\} \subset D$ the following identities hold:

3.10 (i) $x \cap (\bigcup_B y_\beta) = \bigcup_B (x \cap y_\beta)$ and, dually,

(ii) $\bigcup_A x_\alpha \cap \bigcup_B y_\beta = \bigcup_{A,B} (x_\alpha \cap x_\beta)$

Here $(\bigcup_M a_\alpha)$ is supremum of the set $\{a_\alpha | \alpha \in M\}$ and $(\bigcap_M a_\alpha)$ is infimum of the set $\{a_\alpha | \alpha \in M\}$.

3.11 Lemma.

Let D be any finite (and therefore complete) distributive lattice; f and h be any continuous functions from $[D \to D]$. Then $g = \lambda x \in D \,.\, f(x) \cap h(x)$ is continuous.

Proof:

Let $X \subseteq D$ be any directed subset of D.

3.11.1 $g(\cup X) = f(\cup X) \cap h(\cup X)$

by continuity of f and h

$= (\cup f(X)) \cap (\cup h(X))$,

by distributivity of D

$= \cup\{f(x) \cap h(y) | x,y \in X\}$.

On the other hand

3.11.2 $\cup(g(X)) = \cup\{f(x) \cap h(x) | x \in X\}$.

It follows that, since $\cup X \geq X$,

3.11.3 $g(\cup X) \geq \cup\, g(X)$.

However, since X is directed, for any $x,y \in X$ there exists $z \in X$ such that $x,y \leq z$. Therefore $f(x) \leq f(z)$ and $h(y) \leq h(z)$; it follows then that

3.11.4 $f(x) \cap h(y) \leq f(z) \cap h(z)$

This means that the set $A_1 = \{f(x) \cap h(y) | x,y \in X\}$ is majorized by the set $A_2 = \{f(z) \cap h(z) | z \in X\}$, in the sense that for any element a_1 of A_1, we can find an element a_2 of A_2 such that $a_2 \geq a_1$. Hence $\cup A_2 \geq \cup A_1$, or by (3.11.2) and (3.11.1)

3.11.5 $g(\cup X) \leq \cup g(X)$

(3.11.5) and (3.11.3) give us

3.11.6 $g(\cup X) = \cup g(X)$.

That is, g is continuous: q.e.d.

3.11.7 Corollary.

Let f_i: D --> D be a collection of continuous mappings. Define $f = \lambda x. \in D . \bigcap_i f_i(x)$ and D is a finite distributive lattice. Then f is continuous.

Proof: Induction on the number of f_i's using lemma (3.11).

3.11.8 Lemma.

Let D be a finite distributive lattice and $F \subseteq [D \to D]$. Then $(\bigcap F)(x) = \bigcap\{f(x) \mid f \in F\}$ for any $x \in D$.

Proof: Since D is finite, F is finite: i.e., $F = \{f_1, f_2, \ldots, f_k\}$.

Let $g = \lambda x \in D . \bigcap_{i=1}^{k} f_i(x)$. g is continuous by the corollary (3.11.7). Take any x in D. Then

$g(x) = \bigcap_{i=1}^{k} f_i(x) \leq f_i(x)$ for any $1 \leq i \leq k$.

3.11.9 From this $g \leq \bigcap F$.

But if $h \leq f_i$ for any $1 \leq i \leq k$, then

$h(x) \leq f_i(x)$ for any i and $x \in D$. Therefore

$h(x) \leq \bigcap_i f_i(x)$ and hence

3.11.10 $h \leq g$ by extensionality.

From (3.11.9) and (3.11.10) deduce

3.11.11 $g = \bigcap F$.

Hence $(\bigcap F)(x) = g(x) = \bigcap_{i=1}^{k} f_i(x) = \bigcap\{f(x) \mid f \in F\}$: q.e.d.

3.11.12 Lemma.

If D_0 is a finite distributive lattice, then

(i) $\Psi_n(\bigcap X) = \bigcap \Psi_n(X)$

(ii) $\Phi_n(\bigcap Y) = \bigcap \Phi_n(Y)$

for any $n \geq 0$ and $X \subseteq D_{n+1}$, $Y \subseteq D_n$

Proof: We prove it by induction on n.

(1) Let $n = 0$.

We have $\Phi_0(\bigcap Y) = \lambda x \in D_0 . (\bigcap Y)$

by lemma (3.11.8) and extensionality

$$= \bigcap\{\lambda x \in D_0 . y \mid y \in Y\} = \bigcap \Phi_0(Y)$$

By the definition (1.8):

$\Psi_0(\bigcap X) = \bigcap(X)(\bot)$

by lemma (3.11.8)

$= \bigcap\{f(\bot) \mid f \in X\} = \bigcap\{\Psi_0(X)\}$.

(2) Assume (i) and (ii) for all $n \leq k$. Let $x \in D_k$. Then, by (1.8)

$\Psi_{k+1}(\bigcap X)(x) = \Psi_k \circ (\bigcap X) \circ \Phi_k(x)$

lemma (3.11.8)

$= \Psi_k(\bigcap\{f(\Phi_k(x)) \mid f \in X\})$

by the induction hypothesis

$= \bigcap\{\Psi_k \circ f \circ \Phi_k(x) \mid f \in X\}$

by the definition (1.8)

$= \bigcap\{\Psi_{k+1}(f)(x) \mid f \in X\} = \bigcap\{\Psi_{k+1}(X)(x)\}$

by lemma (3.11.8)

$= (\bigcap\{\Psi_{k+1}(X)\})(x)$, which proves the induction hypothesis for k+1 by extensionality. Proof for Φ_{k+1} is analogous to that above: q.e.d.

3.12 *Lemma.*

In any complete distributive lattice D, identities (3.10.i) and (3.10.ii) are equivalent.

3.13 *Lemma.* Due to Birkhoff (1960:147)

A complete lattice D is relatively pseudo-complemented if and only if it satisfies (3.10).

3.14 *Lemma.* Due to Barendregt (1977:1111)

Let D be a complete lattice and A some set. Let

$X = \{\varphi_\alpha | \varphi_\alpha \in [D \rightarrow D] \ \& \ \alpha \in A\}$.

Then $g = (\lambda x. \bigcup_A (\varphi_\alpha(x)))$ is a continuous function: That is, $g \in [D \rightarrow D]$.

From Lemma (3.14) it is an easy exercise to show the following corollary.

3.14.1 *Corollary.*

For any $f, g \in [D \rightarrow D]$, $f \cup g = \lambda x. \in D \ . \ f(x) \cup g(x)$.

3.15 *Definition.*

Let D be any complete and *finite relatively pseudo-complemented lattice* (r.p.c.). As in (1.8) we define D_∞ based on $D_0 \equiv D$.

The following theorem is a first step toward the proof of hypothesis (2.25).

3.16 *Theorem.*

If D_0 is defined as in (3.15), then for any n, D_n is a complete relatively pseudo-complemented lattice.

Proof: We prove the theorem by induction on n.

(i) For $n = 0$, D_0 is r.p.c.

(ii) Assume inductively that for any $n \le k$, the lattice D_n is a complete r.p.c. Let f and h be arbitrary elements of $[D_k \to D_k] = D_{k+1}$. That is, f and $h \in D_{k+1}$. We will show that $(f \Rightarrow h)$ is well defined in D_{k+1}.

3.17 Let $X = \{\varphi_\alpha \in D_{k+1} | f \cap \varphi_\alpha \le h\}$.

Observe that $X \neq \phi$ since $\perp_{D_{k+1}} \in X \subset D_{k+1}$.

Since D_{k+1} is a complete lattice, $\cup X$ exists.

Let $A = \{\alpha | \varphi_\alpha \in X\}$. We will be interested in

3.18 $\cup X = (\bigcup_{\alpha \in A} \{\varphi_\alpha\}) \in D_{k+1}$.

The claim is that $\cup X = (f \Rightarrow h)$, where $(f \Rightarrow h)$ is a pseudo-complement of f relative to h in the lattice D_{k+1}.

By lemma (3.3) $\cup X = (f \Rightarrow h)$ is equivalent to

3.19 (1) $f \cap (\cup X) \le h$

3.20 (2) If $f \cap \varphi \le h$, then $\varphi \le \cup X$, where $\varphi \in D_{k+1}$.

(2) can be deduced immediately from the definition (3.17.1) of the set X. Because D_k is a complete r.p.c. lattice, elements of D_k satisfy equation (3.10) by lemma (3.13). That is,

3.21 for all $y \in D_k$ and $\{y_\beta | \beta \in B\} \subset D_k$

$y \cap (\bigcup_B y_\beta) = \bigcup_B (y \cap y_\beta)$.

To prove (3.19) we will need to show that:

3.22 $(\bigcup_{\alpha \in A} \{\varphi_\alpha\})(x) = \bigcup_{\alpha \in A} \{\varphi_\alpha(x)\}$ for any $x \in D_k$. Here

$S = \{\varphi_\alpha | \alpha \in A\}$ is any subset of D_{k+1}. $\cup S$ exists and belongs to D_{k+1}

by the same arguing as in (3.18). By the definition

3.23 $F = (\bigcup_{\alpha \in A} \{\varphi_\alpha\}) \geq \varphi_\alpha$ for any $\alpha \in A$. Thus

3.24 $F(x) = (\bigcup_{\alpha \in A} \{\varphi_\alpha\})(x) \geq \varphi_\alpha(x)$ for any $\alpha \in A$ and any $x \in D_k$

so $F(x)$ is an upper bound of $\{\varphi_\alpha(x) | \alpha \in A\}$ for any fixed x. Hence

3.25 $F(x) \geq \bigcup_{\alpha \in A} \{\varphi_\alpha(x)\}$ (which is least upper bound).

3.26 Let $g = \lambda x \in D_k \,.\, \bigcup_{\alpha \in A} \{\varphi_\alpha(x)\}$.

The function g is continuous by lemma (3.14);

3.27 Hence $g \in D_{k+1}$, but $g \geq \varphi_\alpha$ for any $\alpha \in A$: i.e., g is an upper bound for the set S. Hence

3.28 $g \geq \bigcup_{\alpha \in A} \{\varphi_\alpha\}$ (which is least upper bound).

Then clearly

3.29 $g(x) \geq (\bigcup_{\alpha \in A} \{\varphi_\alpha\})(x)$ for any $x \in D_k$.

Or, by the definition (3.26):

3.30 $\bigcup_{\alpha \in A} \{\varphi_\alpha(x)\} \geq (\bigcup_{\alpha \in A} \{\varphi_\alpha\})(x)$ for any $x \in D_k$.

But the reverse inequality (3.25) implies:

3.31 $\bigcup_{\alpha \in A} \{\varphi_\alpha(x)\} = (\bigcup_{\alpha \in A} \{\varphi_\alpha\})(x)$ for any $x \in D_k$. That is

3.32 $F = g$.

Now we can continue proving (3.19). By lemma (3.11.8)

3.33 $(f \cap (\cup X))(x) = f(x) \cap (\cup X)(x)$ for any $x \in D_k$,

by (3.18)

$$= f(x) \cap (\bigcup_{\alpha \in A} \{\varphi_\alpha\})(x)$$

Then by 3.31)

$$= f(x) \cap (\bigcup_{\alpha \in A} \{\varphi_\alpha(x)\})$$

and then by the distributivity of D_k, (3.21) and (3.17),

$$= \bigcup_{\alpha \in A} \{f(x) \cap \varphi_\alpha(x)\} \leq \bigcup_{\alpha \in A} (h(x)) = h(x)$$

Therefore, from (3.33) we conclude that

3.34 $f \cap (\cup X) \leq h$,

proving (3.19). And since (3.20) was proven before, we can conclude that

3.35 $\cup X = (f \Rightarrow h) \in D_{k+1}$.

And since f and h are any elements of D_{k+1}, this proves that D_{k+1} is r.p.c. lattice: q.e.d.

This completes the inductive argument and proves the theorem.

Let us prove the converse of theorem (3.16).

3.36 Theorem

Let D_n be r.p.c. lattice for some $n \geq 1$. Then D_k is r.p.c. lattice for any $k \leq n$.

Proof: It is enough to prove that if D_{k+1} is r.p.c., so is D_k for any $k < n$. So we assume that D_{k+1} is a r.p.c. lattice. Let x and y be any two elements of D_k, and let (Φ_k, Ψ_k) be a projection pair as in (1.9).

Hence, $\Phi_k(x)$ and $\Phi_k(y)$ are in D_{k+1}, which is r.p.c.

3.37 Denote $C = (\Phi_k(x) \Rightarrow \Phi_k(y)) \in D_{k+1}$,

then

3.38 $\Psi_k(C) = \Psi_k(\Phi_k(x) \Rightarrow \Phi_k(y)) \in D_k$.

By the definition of relative pseudo-complements,

3.39 $\Phi_k(x) \cap t \le \Phi_k(y)$ iff $t \le C$ for any $t \in D_{k+1}$.

3.40 Let $x \cap d \le y$ in D_k, then, since Φ_n is a distributive continuous projection (lemma (1.9)), it follows, by lemma (3.11.12) that

3.41 $\Phi_k(x) \cap \Phi_k(d) \le \Phi_k(y)$. Then applying (3.39) we deduce

3.42 $\Phi_k(d) \le C$.

Thus, in particular by (1.7), (1.9),

3.43 $\Psi_k \circ \Phi_k(d) \le \Psi_k(C)$

or

3.44 $d \le \Psi_k(C)$.

3.45 This proves that if $x \cap d \le y$, then $d \le \Psi_k(C)$.

On the other hand,

3.46 if $d \le \Psi_k(C)$, then by (1.9)

3.47 $\Phi_k(d) \le \Phi_k \circ \Psi_k(C) \le C$.

Therefore, by (3.39)

3.48 $\Phi_k(x) \cap \Phi_k(d) \le \Phi_k(y)$.

Consequently, by lemma (3.11.12), we have

3.49 $\Psi_k(\Phi_k(x) \cap \Phi_k(d)) \le \Psi_k \circ \Phi_k(y)$ and thus, by lemma (3.11.12) and (1.7), 3.50 $x \cap d \leq y$.

We conclude from (3.45), (3.46), (3.50) that

3.51 $x \cap d \leq y$ iff $d \leq \Psi_k(C)$.

This proves that there exists $(x \Rightarrow y)$ in D_k and that

3.52 $(x \Rightarrow y) = \Psi_k(\phi_k(x) \Rightarrow \Phi_k(y))$.

Since x and y are arbitrary elements of D_k, (3.52) proves that D_k is a r.p.c. lattice, and this concludes the proof of the theorem.

3.53 Corollary.

If there exists n such that D_n is r.p.c., then D_0 is r.p.c. also.

Proof: Immediately proved by 3.36.

3.54 Definition.

For $m \geq n$ define $\Psi_{m,n}$ which is a mapping from D_m onto D_n.

(i) $\Psi_{n,n}$ is an identity function on D_n.

(ii) $\Psi_{n+k+1,n} = \Psi_n \circ \cdots \circ \Psi_{n+k-1} \circ \Psi_{n+k}$.

3.55 Definition.

Assume that D_0 is a complete r.p.c. lattice. Let $x \in D_\infty$ and $y \in D_\infty$: that is, $x = \langle x_n \rangle_{n=0}^{\infty}$ and $y = \langle y_n \rangle_{n=0}^{\infty}$.

One can define a binary operation "$\Rightarrow$" such that

$(x \Rightarrow y) \equiv z$ where $z = \langle z_n \rangle_{n=0}^{\infty}$ and

(i) $z_0 = \bigcap_{k=1}^{\infty} \{(x_0 \Rightarrow y_0), \Psi_{1,0}(x_1 \Rightarrow y_1), \ldots, \Psi_{k,0}(x_k \Rightarrow y_k), \ldots\}$.

(ii) $z_n = \bigcap_{k=0}^{\infty} \{(x_n \Rightarrow y_n), \Psi_{n+1,n}(x_{n+1} \Rightarrow y_{n+1}), \ldots, \Psi_{n+k,n}(x_{n+k} \Rightarrow y_{n+k}), \ldots\}$, where $(x_m \Rightarrow y_m)$ in (i) and (ii) is a pseudo-complement of x_m relative to y_m in the lattice D_m. This pseudo-complement is well defined for any m, since we assume that D_0 is a r.p.c. lattice and therefore D_m is a r.p.c. lattice for any m by theorem (3.16). Also, z_n is well defined since each D_n is a complete lattice.

3.56 Lemma.

$$\Rightarrow : D_\infty^2 \to D_\infty \text{ , i.e., if } x,y \in D_\infty\text{, then}$$

$(x \Rightarrow y) = z \in D_\infty$ also.

Proof: By definition, $z = \langle z_n \rangle_{n=0}^{\infty}$ is an element of D_∞ iff

(i) for any m, $z_m \in D_m$

(ii) for any m, $z_m = \Psi_m(z_{m+1})$.

(i) follows easily from the definitions (3.54) and (3.55), and from the fact that D_m is a complete lattice for any $m \geq 0$. To prove (ii), the following inequality is needed:

3.57 $\forall n \geq 0 \quad \Psi_n(x_{n+1} \Rightarrow y_{n+1}) \leq x_n \Rightarrow y_n$,

which is easily verified as follows:

From the consistency condition for coordinates in D_∞:

3.57.1 $x_n \cap \Psi_n(x_{n+1} \Rightarrow y_{n+1}) = \Psi_n(x_{n+1}) \cap \Psi_n(x_{n+1} \Rightarrow y_{n+1})$

by lemma (3.11.12)

$= \Psi_n(x_{n+1} \cap (x_{n+1} \Rightarrow y_{n+1}))$.

3.57.2 [We have $a \cap (a \Rightarrow b) = a \cap b$ in any r.p.c. lattice [H. Rasiowa and R. Sikorski (1970:60)]

Hence, by (3.11.12),

$= \Psi_n(x_{n+1} \cap y_{n+1}) = \Psi_n(x_{n+1}) \cap \Psi_n(y_{n+1})$

$= x_n \cap y_n \leq y_n.$

So $x_n \cap \Psi_n(x_{n+1} \Rightarrow y_{n+1}) \leq y_n$.

Then by lemma (3.3)

$\Psi_n(x_{n+1} \Rightarrow y_{n+1}) \leq x_n \Rightarrow y_n$, i.e., (3.57).

By the definition of $\bar{z}$ in (3.55), we have

3.58 $\Psi_m(z_{m+1}) = \Psi_m \bigcap_{k=0}^{\infty} \{(x_{m+1} \Rightarrow y_{m+1}),\ldots,\Psi_{m+k,m+1}(x_{m+k} \Rightarrow y_{m+k}),\ldots\}$.

Even though we have $\bigcap_{k=0}^{\infty}$, there is only a finite number of elements in D_{m+1} (since D_0 is finite). Therefore, lemma (3.11.12) can be applied:

$= \bigcap_{k=0}^{\infty} \{\Psi_m(x_{m+1} \Rightarrow y_{m+1}),\ldots, \Psi_m \circ \Psi_{m+k,m+1}(x_{m+k} \Rightarrow y_{m+k}),\ldots\}$.

Using inequality (3.57) we assure that adding a new, greater term does not alter the value of the infimum:

$= \bigcap_{k=0}^{\infty} \{(x_m \Rightarrow y_m), \Psi_m(x_{m+1} \Rightarrow y_{m+1}),\ldots; \Psi_{m+k,m}(x_{m+k} \Rightarrow y_{m+k}),\ldots\}$,

by the definition (3.55)

$= z_m$. That is,

3.59 $\Psi_m(z_{m+1}) = z_m$, which concludes the proof of (ii).

This completes the proof that $(x \Rightarrow y) = z \in D_\infty$ for any x and y in D_∞.

3.60 Theorem.

Let D_0 be a finite relatively pseudo-complemented lattice and "$\Rightarrow$" be a binary operation defined in (3.55). Then D_∞ is a complete relatively pseudo-complemented lattice with operation of the relative pseudo-complementation given by the "$\Rightarrow$".

Proof:

Let $x = \langle x_n \rangle_{n=0}^{\infty}$ and $y = \langle y_n \rangle_{n=0}^{\infty}$ be in D_∞. By the definition of projection and lemma (3.11.12),

3.60.1 $(x \cap (x \Rightarrow y))_n = x_n \cap (x \Rightarrow y)_n$

by the definition (3.55) of $\Rightarrow$

$= x_n \cap \bigcap_{k=0}^{\infty} \{(x_n \Rightarrow y_n), \ldots, \Psi_{n+k,n}(y_{n+k} \Rightarrow y_{n+k}), \ldots\} \leq$

and, using the inequality (3.57), and by (3.57.2), we obtain

$\leq x_n \cap (x_n \Rightarrow y_n) = x_n \cap y_n \leq y_n$ for any $n \geq 0$.

So from the definition of the ordering relation in D it follows that

3.60.2 $x \cap (x \Rightarrow y) \leq y$.

3.60.3 Let $z \in D_\infty$ and $x \cap z \leq y$.

We would like to show that, in this case,

3.60.4 $z \leq (x \Rightarrow y)$, i.e., for any $n \geq 0$, $z_n \leq (x \Rightarrow y)_n$.

We will prove this by induction on n:

(i) $n = 0$: Since $x_k \cap z_k \leq y_k$ for any $k \geq 0$ by assumption (3.60.3). Then, by the definition for r.p.c.

3.60.5 for any $k \geq 0$, $z_k \leq x_k \Rightarrow y_k$ in D_k

(D_k is an r.p.c. lattice by the theorem (3.16)).

Therefore $(x \Rightarrow y)_0 = \bigcap\{x_0 \Rightarrow y_0, \Psi_0(x_1 \Rightarrow y_1), \ldots\}$

by (3.60.5)

$\geq \bigcap\{z_0, \Psi_0(z_1), \ldots\} = z_0$, q.e.d.

(ii) Let, for $n \leq m$, $z_n \leq (x \Rightarrow y)_n$. Then

$$(x \Rightarrow y)_{m+1} = \bigcap\{(x_{m+1} \Rightarrow y_{m+1}), \Psi_{m+1}(x_{m+2} \Rightarrow y_{m+2}), \ldots\}$$

And, by inequality (3.60.5)

$\geq \bigcap\{z_{m+1}, \Psi_{m+1}(z_{m+2}), \ldots\} = z_{m+1}$.

This concludes the inductive argument and proves (3.60.4).

Finally, (3.60.2) and (3.60.4) are precisely the inequalities present in the definition (3.2) of a relative pseudo-complement. Therefore, we conclude that

$(x \Rightarrow y)$ given by (3.55) is a pseudo-complement of x relative to y. Since x and y were arbitrary in D_∞, by the definition (3.4), deduce that D_∞ is an r.p.c. lattice. q.e.d.

This concludes the proof of hypothesis (2.25).

The notion of the relative pseudo-complement and the proofs related to it are not simple, as apparent in the work done above. Can we simplify our hypothesis? Can we assume that D_∞ is a Boolean lattice, say? The following theorem proves that it is not possible to conceive D_∞ as a Boolean algebra.

3.61 Theorem.
D_∞ cannot be constructed to be a Boolean algebra.

Proof: In a Boolean algebra B complementation(*) is defined and it should satisfy the following axiom:

3.61.1 $x^{**} = x$ for any $x \in B$ (this is a double negation axiom).

However, assume that (*) is defined in D_∞ and satisfies axioms of Boolean logic (they can be found in any textbook on logic).

3.61.2 Let $I = (\lambda x \in D_\infty .\ x) \in D_\infty$.

Then, by assumption, $I^* \in D_\infty$ is well defined and satisfies the axiom

3.61.3 $I^* \cup I = \top$

$(I^* \cup I)(\bot) = I^*(\bot) \cup I(\bot) = I^*(\bot)$

by (3.61.3)

$= \top(\bot) = \top$.

3.61.4 i.e., $I^*(\bot) = \top$.

On the other hand, I^* should satisfy the axiom

3.61.5 $I^* \cap I = \bot$.

Therefore $(I^* \cap I)(\top) = I^*(\top) \cap I(\top)$

$= I^*(\top) = \bot(\top) = \bot$ by (3.61.5)

3.61.6 i.e., $I^*(\top) = \bot$.

(3.61.4) and (3.61.6) show that I^* is not a monotonic function and thus is not continuous in D . We conclude that $I^* \notin D_\infty$, which means that the Boolean complementation in D_∞ is not well defined; that is, D_∞ is not a Boolean lattice.

Before we conclude this section, we will present a few results, describing properties of the operation "*" which is pseudo-complementation in D_∞ and was defined in (3.5). Proofs are essentially based on the theorems and lemmas presented throughout section 3 and will appear elsewhere in our works.

3.62 Lemma.

Let D_∞ be constructed as in (3.60) and "*" be an operation of pseudo-complementation in D_∞. Then, for any $f \in D_\infty$ and any $x \in D_\infty$ $f^{**}(x) \geq (f(x))^{**}$, where $f^{**} \equiv (f^*)^* \in D_\infty$.

3.63 Corollary.

$g = \lambda fx.\ (f(x))^{**}$ is a continuous function: i.e., $g \in D_\infty$.

3.64 Lemma.

For any $n \geq 0$ if $x \in D_n^\infty$ and $x^{**} = x$, then $x \in D_0^\infty$.

3.65 Theorem.

If D_0 is a Boolean lattice, then mapping $\varphi: x \longmapsto x^{**}$ is a lattice-homomorphism of D_∞ onto D_0^∞.

Note: This theorem assures us that maximal Boolean subalgebra of D_∞ is contained in D_0^∞. For the reader familiar with lattice theory, theorem (3.65) will resemble Glivenko's theorem (see Birkhoff (1960:148)).

4. λ-LOGIC

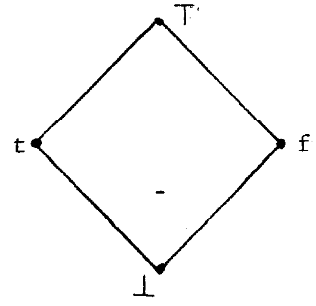
In formalized theories it is a tradition to describe first their language, set of logical axioms and rules of inference. Only after this description is the notion of structure or semantics introduced. Structure (for definition, see any modern textbook on mathematical logic) is usually used for the formulation of nonlogical axioms. We then require that all the theorems of a formalized theory be valid statements about the structure. The existence of a model proves the consistency of the theory. The universe of a structure is postulated to be some nonempty set. Recent developments in the categorical analysis of logic show that this postulate carries its own logical calculus -- namely, classical Boolean logic. That is why for the longest time Boolean logical calculus was considered to be *the correct logic*. Development of the Topoi theory clearly demonstrated that there is no one correct logic, just as there is no one correct geometry. Thus, the choice of *logical* axioms depends on the choice of a structure. If we choose a universe from a category that differs from the category Set , we will conceive a nonclassical "logic" which differs from Boolean logic.

Since our purpose is to find an appropriate logic for λ-calculus, and not to impose some kind of logic onto it, we inevitably digress from the tradition mentioned above. "Model," not "language," is our starting point.

A. *Structure for λ-logic*.

4.1 *Definition*.

Let $D = \{\bot, t, f, \top\}$: That is, the 4-valued lattice:



Then define D_∞ as in (1.8), based on $D_0 = D$. Ordering in D_∞ is denoted $\leq$.

This particular D_∞ was suggested by Dana Scott (1973). The reason for such a choice of D_0 is dictated by the necessity of introducing a continuous conditional, or choice operation. In fact, if we didn't require the conditional to be continuous, our choice would be $D_0 = \{\bot, \top\}$: i.e., the simplest nontrivial Boolean lattice.

4.2 Definition.

The model of λ-logic is the lattice D_∞ of (4.1). By theorem (3.60) D_∞ is a complete relatively pseudo-complemented lattice.

B. The Language of λ-logic.

The language of our formalized theory is an extension of the λ-calculus. Since the model was already defined, it is possible to assign interpretations $[\![\ ]\!]$ in D_∞ concurrently with language definitions.

4.3 Definition.

Let ρ be any valuation in D_∞ (see definition (1.18)). Assume denumerably many variables x,y,z,.... The set of formulas is defined by induction as follows:

1. Each variable is a formula.

If x is a variable

(A1) $[\![x]\!](\rho) = \rho(x)$.

2. If A and B are formulas, so is AB

(A2) $[\![AB]\!](\rho) = \Phi([\![A]\!](\rho))([\![B]\!](\rho))$

3. If A and B are formulas, then (PAB) is a formula and

(A3) $[\![PAB]\!](\rho) = [\![A]\!](\rho) \Rightarrow [\![B]\!](\rho)$.

4. If A is a formula, then so is (NA), and

(A4) $\llbracket NA \rrbracket (\rho) = \llbracket A \rrbracket (\rho) \Rightarrow \bot$, where $\bot$ is the smallest element in D_∞.

5. If A and B are formulas, then so are ($\vee AB$) and ($\wedge AB$), where interpretations are given by

(A5) $\llbracket \vee AB \rrbracket(\rho) = \llbracket A \rrbracket(\rho) \cup \llbracket B \rrbracket(\rho)$;

(A6) $\llbracket \wedge AB \rrbracket(\rho) = \llbracket A \rrbracket(\rho) \cap \llbracket B \rrbracket(\rho)$.

6. If A, B, C are formulas, then Cond(ABC) is a formula with interpretation

$$(A7)\quad \llbracket \mathrm{Cond}(ABC) \rrbracket (\rho) = \begin{cases} \llbracket B \rrbracket(\rho) \text{ if } t \leq \llbracket A \rrbracket(\rho) \text{ and } f \not\leq \llbracket A \rrbracket(\rho) \\ \llbracket C \rrbracket(\rho) \text{ if } t \not\leq \llbracket A \rrbracket(\rho) \text{ and } f \leq \llbracket A \rrbracket(\rho) \\ \llbracket \vee BC \rrbracket(\rho) \text{ if } t \leq \llbracket A \rrbracket(\rho) \text{ and } f \leq \llbracket A \rrbracket(\rho). \\ \bot \text{ otherwise.} \end{cases}$$

This definition of the conditional is Scott's (1973).

<u>Note</u>: We will abbreviate (2), (4), (5), (6) as

(i) $PAB \equiv A \Rightarrow B$

(ii) $NA \equiv \sim A$

(iii) $\vee AB \equiv A \vee B$

(iv) $\wedge AB \equiv A \wedge B$

(v) $\mathrm{Cond}\ (A,B,C) \equiv (A \supset B,C)$.

7. If A and B are formulas, so is (QAB).

$$(A8)\quad \llbracket QAB \rrbracket (\rho) = \begin{cases} t \text{ if } \llbracket A \rrbracket(\rho) = \llbracket B \rrbracket(\rho); \\ f \text{ otherwise.} \end{cases}$$

8. If x is a variable and A is a formula, then the abstraction ($\lambda x.A$) is a formula iff it has the interpretation $\llbracket \lambda x.A \rrbracket(\rho)$ for any ρ, that is

(A9) $\llbracket \lambda x.A \rrbracket = \Psi(\lambda d \in D_\infty . \llbracket A \rrbracket(\rho(d/x)))$,

where $\rho(d/x)$ is defined in (1.18). Note (A9) is well defined iff $\lambda d \in D_\infty . \llbracket A \rrbracket(\rho(d/x))$ is a <u>continuous</u> function on D_∞, e.g.., $\lambda x.Nx$ is <u>not</u> a formula by this definition.

9. L and R are formulas, and

(i) $[\![L]\!](\rho) = t \in D_\infty$;

(ii) $[\![R]\!](\rho) = f \in D_\infty$.

10. T and F are formulas, and

(i) $[\![T]\!](\rho) = \top$: maximal element in D_∞;

(ii) $[\![F]\!](\rho) = \bot$: minimal element in D_∞.

11. If A and B are formulas, then A = B is a formula

$$[\![A = B]\!](\rho) = \begin{cases} \top \text{ if } [\![A]\!](\rho) = [\![B]\!](\rho), \\ \bot \text{ otherwise.} \end{cases}$$

If parentheses are absent, we assume application of formulas from the left to the right. The scope of λx is defined similarly to (1.1).

4.4 Definition.

A variable x's occurrence is free in a formula A if x is not in the scope of λx; otherwise x is bound in A. FV(A) is the set of all free variables in A. A is closed if $FV(A) = \phi$. If A and B are formulas, then $[A/x]B$ is defined as in (1.1).

4.5 Observation.

All λ-terms are formulas.

[illegible] oms and rules of inference.

4.6 Defini[illegible]

Let A be an[illegible] a of λ-logic. Then we say that A is valid iff $[\![A]\!](\rho) = \top$ for any valua[illegible] If formula A is valid, we denote it $\vdash A$. We abbreviate if $\vdash A_1, \& \ldots \vdash A_n$ then $\vdash$ [illegible] $A_1, \ldots, A_n \vdash X$.

4.7 Equational axioms:

Let A, B and C be formulas. Then

4.7.1 $\vdash A = A$

4.7.2 $$\frac{\vdash A = B}{\vdash B = A}$$

4.7.3 $$\frac{\vdash A = B, \vdash B = C}{\vdash A = C}$$

4.7.4 $$\frac{\vdash A = B}{\vdash C[A] = C[B]}$$ for all contexts C[ ].

4.7.5 (α) $\lambda x.A = \lambda y.[y/x]A$, provided $\lambda x.A$ is a formula.

(β) $(\lambda x.A)B = [B/x]A$ if $(\lambda x.A)$ and B are formulas.

(η) $\lambda x.Ax = A$ provided $(\lambda x.Ax)$ is a formula and $x \notin FV(A)$.

4.7.6 Let A be a λ-term, then

$$\frac{\text{A does not have head normal form}}{\vdash A = F}$$

(This postulates the equality of all unsolvable terms. See Wadsworth.)

4.8 (i) $\vdash T$ (iii) $\vdash \lambda x.F = F$ (v) $\vdash \sim T = F$
for any formula A
(ii) $\vdash \lambda x.T = T$ (iv) $\vdash T(A) = T$ (vi) $\vdash F(A) = F$

4.9 Rule Eq. $X = Y$ & $\vdash X$, then $\vdash Y$, where X and Y are formulas.

4.10 Modus ponens. X and Y are formulas, then $X \Rightarrow Y, X \vdash Y$.

4.11 Intuitionistic propositional axioms.

1. $\vdash((A \Rightarrow B) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \Rightarrow C)))$,
2. $\vdash(A \Rightarrow (A \vee B))$,
3. $\vdash(B \Rightarrow (A \vee B))$,
4. $\vdash((A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow ((A \vee B) \Rightarrow C)))$,

5. $\vdash((A \wedge B) \Rightarrow A)$,

6. $\vdash((A \wedge B) \Rightarrow B)$,

7. $\vdash((C \Rightarrow A) \Rightarrow ((C \Rightarrow B) \Rightarrow (C \Rightarrow (A \wedge B))))$,

8. $\vdash((A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \wedge B) \Rightarrow C))$,

9. $\vdash(((A \wedge B) \Rightarrow C) \Rightarrow (A \Rightarrow (B \Rightarrow C)))$,

10. $\vdash((A \wedge {\sim}A) \Rightarrow B)$,

11. $\vdash((A \Rightarrow (A \wedge {\sim}A)) \Rightarrow {\sim}A)$,

where A, B and C are any formulas of λ-logic.

Notice that the axioms in list (4.11) are in the same form as axioms of intuitionistic propositional logic. This replication in form is caused by the fact that models for both theories are r.p.c. lattices.

4.12 <u>W-rule</u>.

Let A and B be λ-terms.

$$\frac{A \underset{W}{\le} B}{\vdash A \Rightarrow B},$$ where $\underset{W}{\le}$ is Wadsworth's ordering relation from (1.23).

4.13

(i) $\vdash L \wedge R = F$

(ii) $\vdash L \vee R = T$

(iii) $\vdash {\sim}L = R$ and $\vdash {\sim}R = L$.

(iv) $\lambda x.L = L$

(v) $\lambda x.R = R$

(vi) $L(A) = L$

(vii) $R(A) = R$, for any formula A.

4.14 <u>COND axiom</u>.

Let A and B be formulas, then

(i) $\vdash \mathrm{Cond}(LAB) = A$

(ii) $\vdash \mathrm{Cond}(RAB) = B$

(iii) $\vdash \mathrm{Cond}((\Delta\Delta)AB) = \Delta\Delta$, where $\Delta = \lambda x.xx$.

4.15

$\vdash A \Rightarrow B, \vdash B \Rightarrow A$ iff $\vdash A = B$

4.16 Q-axiom.

(i) $A = B \vdash QAB = L$

(ii) $A \neq B \vdash QAB = R$

This completes the list of axioms and inference rules for the λ-logic.

4.17 Theorem.

λ-logic is consistent.

Proof:

We prove consistency by showing that axioms (4.7)-(4.16) of the λ-logic are valid in D_∞, i.e., D_∞ is a model of axioms (4.7) - (4.16). Validity of (4.7.1) - (4.7.5) is proved in a way similar to that of theorem (1.20) (see Wadsworth 1976:495). Axiom (4.7.6) is valid (see Wadsworth 1976:514).

(4.8) and (4.9) are clearly valid by the definition of T and by (4.3.11). Validity of modus ponens follows from: Let $\vdash X \Rightarrow Y$ and $\vdash X$: i.e., $[[X]] \leq [[Y]]$ and $[[X]] = \top$, then $[[Y]] = \top$, q.e.d.

Axioms in the list (4.11) are valid since D_∞ is a complete relatively pseudo-complemented lattice (by theorem (3.60)). Then using interpretations (A3), (A4), (A5), (A6) we can easily translate (4.11) into a list of true statements about elements of any Heyting algebra (see Rasiowa and Sikorski 1970).

Validity of (4.12) follows from the theorem (1.24).

(4.13) is proved by showing that t and f are disjoint Boolean elements of D_∞. Proof is done by induction on the structure of elements in D_∞ and is not a difficult exercise for the reader.

Validity of (4.14) follows immediately from (A7). And (4.16) follows from (A8). q.e.d.

The language of λ-logic can be easily extended in order to adjoin existential and universal quantifiers $\underset{x}{\exists}$ and $\underset{x}{\forall}$. Then we can interpret the quantifiers as infinite join and meet operations in D_∞, in a manner similar to that of interpreting ∃ and ∀ in intuitionistic predicate calculus (Rasiowa and Sikorksi 1970). This extension will significantly increase the expressive power of λ-logic. Since D_∞ is a complete lattice, infinite joins and meets always exist.

It is worth it to note that Scott's conditional (A7) can be abstracted: i.e., $\lambda z \, . \, (z \supset x,y)$ is a continuous function in D_∞. This property makes Cond very useful for programming.

CONCLUSION.

N. Bourbaki once said that proofs had to exist before the structure of a proof could be logically analyzed. We hope that our λ-logic is a natural foundation for a large body of writings, known as "fixed point proofs." Since the size of this paper much exceeded the length we had planned on originally, possible applications are not covered here. It might be useful to express so-called Proof Principles (e.g., Scott's Induction Principle) as theorems of λ-logic (extended by quantifiers). Such an exercise does not seem to be difficult, and it can clarify the matter. Maybe it will bring new and useful verification techniques to the fore.

Our approach reflects the view that each language carries its own logic. If so, proof methods should also depend on a programming language (its semantics). This means that so-called "principles" should vary depending on the language. Extending λ-calculi by various δ-rules, and using methods similar to ours, one might detect the logic of this new language.

Another interesting question is, do we know all about the algebraic structure of D_∞? Is it just an r.p.c. lattice, or may D_∞ be a quasi-r.p.c. lattice, which is a model for a system stronger than IL called "constructive logic." Of course, theorem (3.61) shows that the strength of Boolean logic cannot possibly be accessible if one needs higher-order functionals and self-application. But theorem (3.61) does not mean that the relatively weak IL-like λ-logic cannot be improved (e.g., be made into constructive logic).

REFERENCES

Barendregt, Henk. "Open Problems." In λ-Calculus and Computer Science Theory: Proceedings of Symposium Held in Rome, March 1975:25-27.

Barendregt, Henk. "The Type-free Lambda Calculus." In Handbook of Mathematical Logic: Barwise, Jon (Ed.). North Holland: Amsterdam, 1977.

Birkhoff, Garrett. Lattice Theory. American Mathematical Society: Rhode Island, 1960.

Cooper, D.C. "Program scheme equivalences and second order logic." In Machine Intelligence, Vol. 4. Meltzer, B., and Michie, D. (Eds.). Edinburgh U. Press: Edinburgh, 1969:3-15.

Curry, H. B. and R. Feys. Combinatory Logic, Vol. 1. North-Holland: Amsterdam, 1973.

Hindley, R., B. Lercher and J. Seldin. Introduction to Combinatory Logic. Cambridge U. Press: Cambridge, 1972.

Manna, Z. and A. Pnueli. "Formalization of properties of functional programs. JACM 17(3): July 1970:555-569.

Manna, Z. Mathematical Theory of Computation. McGraw-Hill: NY, 1974.

McCarthy, J. "Towards a mathematical science of computation." In Information Processing 1962, Proc. IFIP Congress 62. Popplewell, C. M. (Ed.). North-Holland: Amsterdam, 1963:33-70.

Milner, R. "Logic for Computable Functions: description of a machine implementation." Memo AIM-169: Stanford, 1972.

Milner, R. "Models of LCF." Memo AIM-186: Stanford, 1973.

Morris, James H. "Lambda-Calculus Models of Programming Languages," Ph.D. Thesis, Project MAC. Massachusetts Institute of Technology: Cambridge, MA, 1968.

Rasiowa, Helena and Roman Sikorski. The Mathematics of Metamathematics, 3rd ed.. Polish Scientific Publishers: Warsaw, 1970.

Russell, Bertrand. "Letter to Frege (1906)." In From Frege to Godel, A Source Book in Mathematical Logic, 1879-1931. Jean van Heijenoort (Ed.). Harvard University Press: Cambridge, MA, 1967:124-125.

Russell, Bertrand. "Mathematical logic as based on the theory of types (1908)." In From Frege to Godel, A Source Book in Mathematical Logic, 1879-1931. Jean van Heijenoort (Ed.). Harvard University Press: Cambridge, MA, 1967:150-182.

Scott, Dana. "Models for various type-free calculi," Proc. 4th International Congress for Logic, Methodology, and the Philosophy of Science (Bucharest). North-Holland: Amsterdam, 1973.

Scott, Dana. "Combinators and Classes." In λ-Calculus and Computer Science Theory: Proceedings of Symposium Held in Rome, March 1975.

Scott, Dana. "Logic and Programming Languages. 1976 ACM Turing Award Lecture. In Communications of the ACM 20, No. 9 (September 1977).

Stoy, Joseph E. The Scott-Strachey Approach to Programming Language Theory. The MIT Press: Cambridge, MA, 1977.

Von Neumann, John. "An axiomatization of set theory (1925." In From Frege to Godel, A Source Book in Mathematical Logic, 1879-1931. Jean van Heijenoort (Ed.). Harvard University Press: Cambridge, MA, 1967:393-413.

Wadsworth, C. P. "The relation between lambda-expressions and their denotations." SIAM J. Comput. 5 (3):1967:488-521.

Zermelo, Ernst. "Investigations in the foundations of set theory I." In From Frege to Godel, A Source Book in Mathematical Logic, 1879-1931. Jean van Heijenoort (Ed.). Harvard University Press: Cambridge, MA, 1967:199-215.