MODELS FOR INTUITIONISTIC LOGIC

(With special reference to
'Martin-Löf Type Theory')

by

Howard Cuckle

A thesis submitted to the
University of Oxford
in partial fulfilment of the
requirements for the degree of
Doctor of Philosophy

BODLEIAN LIBRARY OXFORD
DEPOSITED
THESIS
24·VIII·75

Hilary Term, 1975

Balliol College
Oxford

ABSTRACT

In this work I develop a formalization (M-L) of Martin-Löf type theory, the main concern being an accurate definition of what it is to be a model of M-L. Using this definition, I proceed with actual models of M-L (mainly realizability models) to establish the relative consistency of many intuitionistic principles. In addition to their consistency I investigate their interrelationship. The strongest principles which are shown to be consistent with M-L are Church's Thesis and the Fan Theorem.

The expressive power of M-L is used to formalize certain theories. The Theory of Real Numbers, an Intuitionistic Set Theory and Category Theory are all formalizable. The constructions of the latter are used to describe Kripke Models of M-L.

Finally, I prove that the subsystem of M-L obtained by dropping the rules for cartesian proeucts of types and without rules for universes of types has proof theoretic ordinal ω^{ω}.

ACKNOWLEDGEMENTS

This research was undertaken with the support of a State Studentship of the Department of Education and Science.

I would like to thank Dr. R. O. Gandy for the help and encouragement that he has given me throughout this work. I am also indebted to Professor P. Martin-Löf for his suggestions and to Peter Hancock for numerous philosophical discussions pertaining to the foundations of mathematics.

As regards non-mathematical thinkers, I have been most stimulated by these words of John Locke: '... it is ambition enough to be employed as an underlabourer in clearing ground a little, and removing some of the rubbish that lies in the way of knowledge ...' ('An essay Concerning Human Understanding').

CONTENTS

Chapter		Page
	INTRODUCTION	1
1	MARTIN-LÖF TYPE THEORY	3
	§1. The formal system, M-L	5
	§2. A simplified formalization	12
	§3. Some metatheorems of M-L	17
	§4. Some theorems of M-L	19
	§5. Some useful definitions	21
2	MODELS OF M-L	26
	§1. The formal definition	27
	§2. The model of normal terms	28
	§3. Realizability models	29
	§4. Some examples of realizability models	34
	§5. The proof of §1	39
3	RELATIVE CONSISTENCY	46
	§1. Candidates for relative consistency (by realizability)	46
	§2. Other models over an r.s.	53
	§3. Two candidates for consistency (By submodels) ...	57
	§4. Generalization of results over the type structure ...	70
	§5. Stronger continuity principles	73
4	OTHER THEORIES FORMALIZED	80
	§1. The theory of real numbers	80
	§2. Set theory	83
	§3. Category theory	90
5	THE ORDINAL OF M-L	99
	§1. Finitist type theory (ftt)	99
	§2. The ordinal of ftt	117
	§3. The ordinal of M-L	127
	REFERENCES	128

INTRODUCTION

In a series of papers [1972-4], Martin-Löf has developed a theory of types, both informally and as a formal system. That is, at the informal level it is a theory of types in that it elucidates a certain notion of type, and at the formal level it is a typed formalization of intuitionistic mathematics.

Every mathematical object has its own mode of construction; in so far as every mathematical object is a construction it is constructed in a certain way. Because of this, many objects will have a common property; the property that they have a similar mode of construction. It is the common property we refer to when we say that objects are of the same type. So the notion of type we shall be dealing with does not treat types as pre-existing objects but as concepts, which come into existence at the same time as the objects that fall under them. To use Wittgenstein's term they are formal concepts. And he writes 'When something falls under a formal concept as one of its objects, this cannot be expressed by a proposition. Instead it is shown in the very sign for this object' [1921], §4.126.

The type of an object is apparent from its very construction and if we are to have signs for mathematical objects then those signs must show the type of their construction. So if we are to have a formalization of Intuitionistic Mathematics, it must be a typed theory.

Now, if we take the intuitionist position about proofs seriously, a proof is a mathematical object. What makes the modes of construction of various proofs similar, is that they are proofs of the same proposition. In other words we can say that the type of a proof is the proposition of which it is a proof. Again, whilst the types are concepts, as abstractions, they might be treated as mathematical objects. And as objects they will have a mode of construction, a type. We call a type of types, a universe. Hence, a suitable formalization of intuitionistic mathematics will be also a

formalization of Intuitionistic Logic and a Theory of Types.

We begin this work by describing a formal system (M-L) which is a formal version of Martin-Löf type theory. This will be a typed theory suitable for the formalization of Intuitionistic Mathematics. By developing, in chapter 2, what it is to be a model of M-L, the system M-L will itself be an intuitionistic framework for models of Intuitionistic Mathematics. Constructing certain models within this framework we shall establish many relative consistency results for M-L. In chapter 4, the expressive power of M-L is used and certain theories are formalized. Finally, we investigate the proof theoretic strength of an interesting subsystem of M-L, which reflects on the strength of the whole system.

Chapter 1

MARTIN-LÖF TYPE THEORY

A formal version of Martin-Löf type theory begins with a language of typed formal expressions, the kind of typestructure depending on the needs of the formalization. We shall say that a type has been established when we have introduced signs for every object of that type (terms), and a sign for the type itself (a typesymbol). The typesymbol will form part of the terms which stand for objects of that type; that an object is a member of that type will be shown by the term, which is a sign of that object. To establish the type of natural numbers we will introduce a typesymbol N and terms $0^N, 1^N, \ldots$ At the same time the type of natural numbers is an object of some universe of types and so the sign N will also be a term and be typed by the typesymbol for that universe. The universe to which the natural numbers belong is also an object of some universe and so we shall need a whole hierarchy of universe typesymbols: $V_o, V_1, V_2, \ldots$ So the term for the natural numbers will be N^{V_o} and that for the universe of the natural numbers will be $V_o^{V_1}$, etc.

There will not be signs for atomic propositions, as such, but whenever we have established a type we will be able to establish the proposition that two objects of that type are identical.

The more complex typestructure will be constructed by establishing the cartesian product of a family of types over a type and also the disjoint union of a family over a type. To this end we require signs for families of objects and families of types. These signs will be n-ary function constants; and attached to each constant will be the typesymbols of the index types and the typesymbol of the universe to which the members of the family belong.

Suppose that we have established a type with typesymbol A and a family of types with function constant B^{A,V_n} where the family of types are of the nth. universe. Then, we can establish the cartesian product of the family over the index type, by having a typesymbol forming operation Π, giving the type-

symbol '$\Pi(A,B)$'. At the same time we need a term forming operation λ so that if we have introduced a function constant $b^{A,B}$ then we can form the term λb which is a sign for the function from A to B, in the cartesian product, which takes a to b_a. In general, from a n-ary function constant 'B' with index typesymbols 'A_1', 'A_2',..., 'A_n' we can form the (n-1)-ary function constant '$\Pi(A,B)$' with index typesymbols 'A_1','A_2',...,'A_{n-1}' and is a sign for an n-1 indexed family of cartesian products.

In a similar way we can establish the disjoint union of the family B over A, the typical objects being ordered pairs, the first place of which are objects a of A and the second of B_a.

An n-ary function constant is introduced by induction over its index types. When a type has been established we will have signs for every object of that type, and this fact sanctions the induction over it. For example we could introduce the 1-ary function constant $B^{N,V}$ by induction over the natural numbers, as follows:

B_0 is N

B_{n+1} is the type of functions from B_n to N

and then '$\Pi(N,B)$' is the typesymbol, of a transfinite type.

Similarly, we could introduce the 1-ary function constant $b^{N,B}$ by:

b_o is 0

b_{n+1} is $\lambda 0$

and then λb is the function of type $\Pi(N,B)$ which takes a to b_a.

We can think of the induction as a scheme of definition for the introduced constant, then for example, B_0 and N are definitially equal.

The most important feature of a formalization of Martin-Lof type theory as I have, informally, described it above, is that each expression is constructed in such a way that it is made up of its subexpressions as syntactical

parts. This allows us, when constructing models of the formal theory, to interpret a term as a function of the interpretations of its subterms.

§1. The formal system, M-L

We begin by describing the formal expressions of the language without regard for their being terms which are typesymbols or not.

Then quite separately we formalize the relations 'being a member of a type' and 'being definitially equal to'. That is, we make explicit the typestructure by defining the metamathematical relations '... is a term of typesymbol ...' and 'the term ... converts to the term ...'. Bearing in mind Wittgenstein's above quoted words, we shall 'say' what is 'shown' in the signs themselves.

A theorem of M-L is a typesymbol, for which there is a closed term which we can prove is a term of that typesymbol.

1.1.1. The language. We start with the following signs

i) a collection of variables $x,y,z,\ldots\ldots$

ii) for each n, a collection of n-ary function constants

$f,g,h,\ldots\ldots$ $F,G,H,\ldots\ldots$

iii) the auxiliary signs Π,Σ, =, λ, ap, s, rec, pr, un, r, id.

iv) some distinguished 0-ary function constants

$0,\mathbb{N},V_0,V_1,V_2,\ldots\ldots$

v) (,).

And a string is a concatenation of some of these signs, as in the following definition

a) a variable is a string

b) an 0-ary function constant is a string

c) if f is an n-ary function constant and $\sigma_1,.....,\sigma_n$ are strings, then $(f;\sigma_1,.....,\sigma_n)$ is a string.

Next perform an association of signs with strings and signs

i) associate with each string σ the variables

$$x^\sigma, y^\sigma, z^\sigma, ...$$

ii) associate with N the constant 0, associate with V_o the constant N and associate with V_{n+1} the constant V_n

iii) associate with the sequence of strings

$$\sigma_1,\sigma_2,.....,\sigma_n,\sigma_{n+1} \quad \text{and}$$

a) the auxiliary sign Π and two other strings $\sigma_{n+2},\sigma_{n+3}$ the n-ary function constant

$$F^{\sigma_1,...,\sigma_n,\sigma_{n+1},\Pi,\sigma_{n+2},\sigma_{n+3}}$$

It will be more suggestive to write this F as

$$\Pi(\sigma_{n+2},\sigma_{n+3})$$

and to say that it is associated with index/value strings

$$\sigma_1,\sigma_2,.....,\sigma_n/\sigma_{n+1}$$

b)c) similarly for the auxiliary signs Σ and =.

d) the auxiliary sign λ and one other string σ_{n+2}, the n-ary function constant

$$f^{\sigma_1,...,\sigma_n,\sigma_{n+1},\lambda,\sigma_{n+2}}$$

and we suggestively write this f as

$$\lambda\sigma_{n+2}$$

and say that it is associated with index/value strings

$$\sigma_1,\sigma_2,...,\sigma_n/\sigma_{n+1}$$

e)f) similarly for auxiliary signs un and id.

h) the auxiliary sign ap, the n-ary function constant

$$f^{\sigma_1,\ldots,\sigma_n,\ \sigma_{n+1},ap}$$

which we can write as just ap

j)k)l) similarly for the auxiliary signs s, pr, and r.

m) and the auxiliary sign rec, and two other strings $\sigma_{n+2},\sigma_{n+3}$ the n-ary function constant

$$f^{\sigma_1,\ldots,\sigma_n,\sigma_{n+1},rec,\sigma_{n+2},\sigma_{n+3}}$$

and as before we can write this as

$$rec(\sigma_{n+2},\sigma_{n+3})$$

iv) if f is an n-ary function constant associated with the index/value strings $\sigma_1,\sigma_2,\ldots,\sigma_n/\sigma_{n+1}$ and the strings $\tau_1,\ldots,\tau_n$ are associated with $\sigma_1,\ldots,\sigma_n$ respectively, then $(f;\tau_1,\ldots,\tau_n)$ is associated with σ_{n+1}.

Finally we define the formal expressions of M-L

a) $0,N,V_o,V_1,V_2,\ldots\ldots$ are formal expressions

b) if a variable x is associated with a formal expression then it is a formal expression

c) if an n-ary function constant f is associated with index/value strings which are formal expressions and $a_1,\ldots,a_n$ are formal expressions associated with the indexes then $(f;a_1,\ldots,a_n)$ is a formal expression.

1.1.2. The (metamathematical) relations $\in$ and conv.

These are to be read '... is a term of typesymbol ...' and 'the term ... converts to the term ...'. They are defined between the formal expressions of M-L and at the same time as other (metamathematical) predicates (underlined in the definition) as follows:

i) A is a typesymbol if $A \in V_n$ for some n

ii) a is a term if $a \in A$; and hence a term of typesymbol A if A is a typesymbol

iii) the term a <u>depends</u> on the variables its subterms depend on, in addition to those that its typesymbol depends on. We write $a[x_1,\ldots,x_n]$ when a term depends on the $x_1,\ldots,x_n$ and for $1 \leq i \leq n$ x_i depends on only $x_1,\ldots,x_{i-1}$. And $a[b_1,\ldots,b_n]$ as the result of replacing every occurrence of x_1 by $b_1,\ldots,x_n$ by b_n. See footnote , p.25.

iv) if x is a variable associated with the formal expression A and A is a typesymbol, then we say that x is a variable of typesymbol A.

v) an n-ary function constant f <u>is introduced</u> if it is introduced in 1,2,3,6,7,9,10,11,13,14,15 below.

If f is associated with the index/value expression $A_1,\ldots,A_n/A_{n+1}$ and they are typesymbols, then we say that f is introduced with index/value typesymbols $A_1,\ldots A_n/A_{n+1}$.

The remaining clauses of the definition are grouped

<u>Group A</u>

If the variable x is a variable of typesymbol A then:

$$x \in A$$

If the n-ary function constant f is introduced with index/value typesymbols $A_1, A_2[x_1], A_1[x_1,x_2],\ldots, A_n[x_1,\ldots,x_{n-1}]/A[x_1,\ldots,x_n]$ where for $1 \leq i \leq n$, $x_i \in A_i[x_1,\ldots,x_{i-1}]$, then:

$$\frac{a_1 \in A_1, a_2 \in A_2[a_1],\ldots,a_n \in A_n[a_1,\ldots,a_{n-1}]}{(f;a_1,\ldots,a_n) \in A[a_1,\ldots,a_n]}$$

And if $b_1 \in A_1, b_2 \in A_2[b_1],\ldots,b_n \in A_n[b_1,\ldots,b_{n-1}]$ then:

$$\frac{a_1 \text{ conv } b_1,\ a_2 \text{ conv } b_2,\ldots,\ a_n \text{ conv } b_n}{(f;a_1,\ldots,a_n) \text{ conv } (f;b_1,\ldots,b_n)}$$

Also the more straightforward clauses:

$$V_n \in V_{n+1} \qquad \frac{a \in A,\ A \text{ conv } B}{a \in B}$$

$$\frac{a \in A}{a \text{ conv } a} \qquad \frac{a \text{ conv } b}{b \text{ conv } a} \qquad \frac{a \text{ conv } b,\ b \text{ conv } c}{a \text{ conv } c}$$

In what follows we assume that $A_1, A_2[x_1], \ldots, A_n[x_1, \ldots, x_{n-1}]$ are typesymbols and that for $1 \leq i \leq n$, $x_i \in A_i[x_1, \ldots, x_{i-1}]$.

Group B

1) Π- reflection. If $A[x_1, \ldots, x_n] \in V_k$ & $B[x_1, \ldots, x_n, x] \in V_\ell$ where $x \in A[x_1, \ldots, x_n]$ then introduce the n-ary function constant:

$$\Pi(A[x_1, \ldots, x_n],\ B[x_1, \ldots, x_n, x])$$

with index/value typesymbols:

$$A_1, A_2[x_1], \ldots, A_n[x_1, \ldots, x_{n-1}]/V_{\max(k,\ell)}$$

2) Π-introduction. If $b[x_1, \ldots, x_n, x] \in B[x_1, \ldots, x_n, x]$ then introduce the n-ary function constant:

$$\lambda b[x_1, \ldots, x_n, x]$$

with index/value typesymbols :

$$A_1, A_2[x_1], \ldots, A_n[x_1, \ldots, x_{n-1}]/(\Pi(A,B);\ x_1, x_2, \ldots, x_n)$$

3) Π-elimination. Introduce the n+2-ary function constant:

$$ap$$

with index/value typesymbols:

$$A_1, \ldots, A_n, (\Pi(A,B);\ x_1, \ldots, x_n), A[x_1, \ldots, x_n]/B[x_1, \ldots, x_n, x]$$

4) Π-conversion. If $a_1 \in A_1, \ldots, a_n \in A_n[a_1, \ldots, a_{n-1}]$, $a \in A[a_1, \ldots, a_n]$ then

$$(ap; a_1, \ldots, a_n, (\lambda b[x_1, \ldots, x_n, x];\ a_1, \ldots, a_n), a)$$

$$\text{conv } b[a_1, \ldots, a_n, a]$$

Group C

5) N-reflection. $N \in V_o$

6) N-introduction. $0 \in N$

And introduce the 1-ary function constant s, with index/value type-symbols N/N.

7) N-elimination. If for $x \in N$ and $y \in A[x_1,\ldots,x_n,x]$

$$a[x_1,\ldots,x_n] \in A[x_1,\ldots,x_n,0] \;\&\; b[x_1,\ldots,x_n,x,y] \in A[x_1,\ldots,x_n,(s;x)]$$

then introduce the n+1-ary function constant:

$$rec(a[x_1,\ldots,x_n],\ b[x_1,\ldots,x_n,x,y])$$

with index/value typesymbols:

$$A_1,\ldots,A_n[x_1,\ldots,x_{n-1}],N/A[x_1,\ldots,x_n,z] \text{ where } z \in N$$

8) N-conversion. If $a_1 \in A_1,\ldots,a_n \in A_n[a_1,\ldots,a_{n-1}]$, $c \in N$ then

$$(rec(a,b);\ a_1,\ldots,a_n,0)\ conv\ a[a_1,\ldots,a_n]$$

$$(rec(a,b);\ a_1,\ldots a_n,(s;c))\ conv\ b[a_1,\ldots,a_n,c,(rec;a_1,\ldots,a_n,c)]$$

where a and b are as in the hypothesis of 7)

Group D

$$a = a[x_1,\ldots,x_n];\quad b = b[x_1,\ldots,x_n,x,y]$$

9) Σ-reflection. If $A[x_1,\ldots,x_n] \in V_k$ & $B[x_1,\ldots,x_n,x] \in V_\ell$ where $x \in A[x_1,\ldots,x_n]$ then introduce the n-ary function constant:

$$\Sigma(A[x_1,\ldots,x_n],B[x_1,\ldots,x_n,x])$$

with index/value typesymbols:

$$A_1,A_2[x_1],\ldots,\ A_n[x_1,x_n,\ldots,x_{n-1}]/V_{max\ (k,\ell)}$$

10) Σ-introduction. Introduce the n+2-ary function constant pr with index/value typesymbols:

$$A_1,\ldots,A_n[x_1,\ldots,x_{n1}],\ A[x_1,\ldots,x_n],\ B[x_1,\ldots,x_n,x]/(\Sigma(A,B);x_1,\ldots x_n)$$

11) Σ-elimination. If for $x \in A[x_1,\ldots,x]_n$ & $y \in B[x_1,\ldots,x_n,x]$ $c[x_1,\ldots,x_n,x,y] \in C[x_1,\ldots,x_n,(pr;x_1,\ldots,x_n,x,y)]$ then introduce the n+1-ary function constant $un(c[x_1,\ldots,x_n,x,y])$ with index/value typesymbols:

$$A_1,\ldots,A_n,(\Sigma(A,B);\ x_1,\ldots,x_n)/C[x_1,\ldots,x_n,z]$$

where $z \in (\Sigma(A,B);x_1,\ldots,x_n)$

12) Σ-elimination. If $a_1 \in A_1, \ldots, a_n \in A_n, d \in A[a_1,\ldots,a_n]$, $e \in B[a_1,\ldots,a_n,d]$ then $(\mathrm{un}(c);a_1,\ldots,a_n,d,e)$ conv $c[a_1,\ldots,a_n,d,e]$

where c is as in the hypothesis of 11); $c = c[x_1,\ldots,x_n,x,y]$

Group E

13) =-reflection. If $A[x_1,\ldots,x_n] \in V_k$ then introduce the n+2-ary function constant =, with index/value typesymbols:

$$A_1,\ldots,A_n[x_1,\ldots,x_{n-1}],\ A[x_1,\ldots,x_n],\ A[x_1,\ldots,x_n]/V_k$$

14) =-introduction. Introduce the n+1-ary function constant r with index/value typesymbols:

$$A_1,\ldots,A_n[x_1,\ldots,x_{n-1}],\ A[x_1,\ldots,x_n]/(=;x_1,\ldots,x_n,x,x)$$

where $x \in A[x_1,\ldots,x_n]$

15) =-elimination. If $b[x_1,\ldots,x_n,x] \in C[x_1,\ldots,x_n,x,x,(r;x_1,\ldots,x_n,x)]$ then introduce the n+3-ary function constant $\mathrm{id}(b[x_1,\ldots,x_n,x])$ with index/value typesymbols:

$$A_1,\ldots,A_n,\ A[x_1,\ldots,x_n],\ (=;x_1,\ldots,x_n,y,z)/C[x_1,\ldots,x_n,y,z,t]$$

16) =-conversion

$$(\mathrm{id}(b);a_1,\ldots,a_n,c,c,(r;c))\ \mathrm{conv}\ b[a_1,\ldots,a_n,c]$$

where b is as in the hypothesis of 15); $b = b[x_1,\ldots,x_n,x]$

1.1.2.1. Remarks It can be seen that a term $c[x]$ is either a variable or there is a unique (up to change of bound variables) n-ary function constant f so that:

$$c[x] = (f;b_1[x],b_2[x],\ldots,b_n[x])$$

and then substitution is given by; (see p. 25):

$$c[a] = (f;b_1[a],b_2[a],\ldots,b_n[a])$$

We now give an example to show how terms would in practice be formed from a function constant via the rule of term formation.

$A_1 =_{df} N$

$N \in V_o$ and for $X \in V_o$, $N \times X \in V_o$, so introduce the 1-ary function constant:

$$\mathrm{rec}(N, N \times X)$$

with index/value typesymbols:

$$A \ /V_o$$

$A_2[x] =_{df} (\mathrm{rec}(N, N\times X);x)$ where $x \in A_1$

$A_2[x] \in V_o$ so introduce the 3-ary function constant =, with index/value typesymbols:

$$A_1, A_2[x], A_2[x]/V_o$$

$A_3[x,y] =_{df} (=;x,y,y)$ where $y \in A_2[x]$

$0 \in A_2[0]$ conv N and for $z \in N$ and $t \in A_2[z]$; $(pr;0,t) \in A_2[(s;z)]$ conv $N \times A_2[z]$ so introduce the 1-ary function constant:

$$\mathrm{rec}(0,(pr;0,t))$$

with index/value typesymbols:

$$A_1/A_2[x]$$

$a_2[x] =_{df} (\mathrm{rec}(0,(pr;0,t));x)$

Introduce the 2-ary function constant r with index/value typesymbols:

$$A_1, A_2[x]/(=;x,y,y)$$

Now we show how to apply the rule, for this r:

$$\frac{\dfrac{a_1 \in A_1}{(\mathrm{rec}(0,(pr;0,t));a_1) \in A_2[a_1]} \qquad a_1 \in A_1}{(r;a_1, a_2[a_1], a_2[a_1]) \in A[a_1, a_2[a_1]]} \; =_{df} a_2[a_1]$$

Note that a_1 and $a_2[a_1]$ may be open. If $a_1 = 1$ then $a_2[a_1] = (pr;0,0)$ but if $a_1 = b[z]$ then $a_2[a_1] = a_2'[z]$ and $A_2[a_1] = A_2'[z]$ etc.

§2. A simplified formalization

Earlier formalizations did not make use of function constants. Families of types indexed by a type A where represented by typesymbols 'B[x]' where x is a variable of the typesymbol for A. And similarly for families of objects of that family of types, the terms $b[x] \in B[x]$ was used.

Instead of introducing whole families of terms, in the form of a function constant, the rules allowed the definition of an open term of the appropriate typesymbol, whose free variable was of the typesymbol over which the definition takes place.

This allowed us to use more suggestive notation and write the rules in tree form. And as a heuristic device I present a version of it now. The presentation gives the cases of no extra free variables, but they may occur in general throughout the scheme.

Group A

$$\frac{a \text{ conv } b}{c[a] \text{ conv } c[b]} \qquad \text{where for } x \in A,\ c[x] \in C[x] \text{ and } a, b \in A$$

$$V_n \in V_{n+1} \qquad \frac{a \in A \qquad A \text{ conv } B}{a \in B}$$

$$\frac{a \in A}{a \text{ conv } a} \qquad \frac{a \text{ conv } b}{b \text{ conv } a} \qquad \frac{a \text{ conv } b \qquad b \text{ conv } c}{a \text{ conv } c}$$

Group B

Π-reflection

$$\frac{A \in V_m \qquad \begin{array}{c} x \in A \\ \vdots \\ B[x] \in V_n \end{array}}{\Pi x \in A.B[x] \in V_{\max(m,n)}}$$

Π-introduction

$$\frac{\begin{array}{c} x \in A \\ \vdots \\ b[x] \in B[x] \end{array}}{\lambda x b[x] \in \Pi x \in A.B[x]}$$

Π-elimination

$$\frac{f \in \Pi x \in A.B[x] \qquad a \in A}{f(a) \in B[a]}$$

Π-conversion

$$\lambda x b[x](a) \text{ conv } b[a]$$

Group C

N-reflection

$$N \in V_o$$

N-introduction

$$\frac{x \in N}{0 \in N} \qquad \frac{x \in N}{s(x) \in N}$$

N-elimination

$$\frac{z \in N \qquad a \in C[o] \qquad \begin{array}{c} x \in N \quad y \in C[x] \\ \vdots \\ b[x,y] \in C[s(x)] \end{array}}{rec_{a,b}[z] \in C[z]}$$

N-conversion

$$rec_{a,b}(0) \text{ conv } a;\ rec_{a,b}(s(c)) \text{ conv } b[c,rec(c)]$$

Group D

Σ-reflection

$$\frac{A \in V_m \qquad \begin{array}{c} x \in A \\ \vdots \\ B[x] \in V_n \end{array}}{\Pi x \in A.B[x] \in V_{max(m,n)}}$$

Σ-introduction

$$\frac{a \in A \qquad b \in B[a]}{(a,b) \in \Sigma x \in A.B[x]}$$

Σ-elimination

$$\frac{z \in \Sigma x \in A.B[x] \qquad \begin{array}{c} x \in A, \quad y \in B[x] \\ \vdots \qquad\quad \vdots \\ c[x,y] \in C[(x,y)] \end{array}}{un_c[z] \in C[z]}$$

Σ-conversion

$$un_c((a,b)) \text{ conv } c[a,b]$$

Group E

=-reflection

$$\frac{a,b \in A \qquad A \in V_n}{a =_A b \in V_n}$$

=-introduction

$$\frac{a \in A}{r(a) \in a =_A a}$$

=-elimination

$$\frac{z \in a =_A c \qquad \begin{array}{c} t \in A \\ \vdots \\ b[t] \in C[t,t,r(t)] \end{array}}{id_b[z] \in C[a,c,z]}$$

=-conversion

$$id_b(r(a)) \text{ conv } b[a]$$

1.2.1. Remarks on the simplification

a) If $B[x]$ is the same typesymbol B for each $x \in A$ (i.e. B is not dependent on x) we shall use the suggestive abbreviations of $\Pi x \in A.B[x]$ to $A \to B$; and of $\Sigma x \in A.B[x]$ to $A \times B$. Also we abbreviate $\Pi x \in A.\ \Pi y \in A.B[x,y]$ to $\Pi x,y \in A.B[x,y]$; and $(A \to B) \times (B \to A)$ to $A \leftrightarrow B$. And to assist reading we shall assume the convention that $\to$ is the strongest connective; and use dots where necessary to indicate the range of Π and Σ.

b) The elimination and conversion rules should be taken together and thought of as a scheme for defining a new term. In fact when the relevant typesymbols are clear, this can be written, as for example in N-elimination/conversion

$$rec_{a,b}(0) =_{df} a$$

$$rec_{a,b}(s(x)) =_{df} b[x,rec(x)]$$

c) As presented, without extra free variables, the rules are just the no parameter cases of M-L.

d) This formalization, it should be emphasized, is to be taken simply heuristically. Its deficiency is that there are many terms c with subterms a,which are not the result of substituting a for the variable x in an open term c[x]. To see this consider the type B[x]given by the scheme:

$$B[0] =_{df} N$$

$$B[s(y)] =_{df} B[y] \to B[y]$$

then for $f \in \Pi x \in N.B[x]$

f(1)(0) is one such term

For example f(1)(0) is not the result of substituting 1 for x in the term f(x)(0). For the latter is not a term. There may be some other open term c[x] such that c[1] is f(1)(0); but in general, it is undecidable if such a term exists.

The significance of this fact is model theoretic. To model the main rule of convertability; writing $\bar{a}$ for the interpretation of a term a, we would have to have:

$$\frac{\bar{a} \text{ conv } \bar{b}}{\overline{c[a]} \text{ conv } \overline{c[b]}}$$

And to achieve this we must have $\overline{c[a]}$ as the result of substituting $\bar{a}$ for x in an open term $\bar{c}[x]$. We know from what was said above that this is in general impossible. As will be seen in chapter 2 the present formalization overcomes this earlier difficulty; for the interpretation of a term is automatically a function of the interpretations of its subterms.

However, there are neither more nor less theorems using the simpli-

fied formalization and I shall continue to use it heuristically in what follows. It is only where model theoretic interests dominate that I shall revert to the rigour of M-L proper.

§3 Some Metatheorems of M-L

1.3.1. The relations $\in$ and conv are recursive and hence decidable.

1.3.2. Every number theoretic function definable in M-L is recursive

1.3.3. The syntax allows us to 'read off' from a term its typesymbol, hence:

$$a \in A \;\&\; a \in B \Rightarrow A \text{ conv } B$$

$$a \in A \Rightarrow \exists n.\ A \in V_n$$

1.3.4. From the normalization theorem:

$$a \text{ conv } b,\ a \in A,\ b \in B \Rightarrow A \text{ conv } B$$

1.3.5. By induction on the length of conv:

$$a \text{ conv } b \Rightarrow \exists A,B.\ a \in A \;\&\; b \in B$$

1.3.6. $V_n \in V_n$ is inconsistent (Girard [1972])

Also note about the universes, that the n in V_n is not a variable, it is metamathematical. And that we do not have induction on the universes as they are open ended.

1.3.7. M-L is sound with respect to the intended interpretation. By way of proof we shall just give the intended interpretation. The syntax does not make the distinction between typesymbols which are intended to be signs for types which are not propositions and those that are. Nevertheless, it is clear in particular cases which is intended. For example, N is the former and a=b, the latter.

Call a family of propositions a predicate or property of the index type. Then making the impossible syntactical distinctions we could write:

$\overline{P \to Q}$ = $\bar{P} \supset \bar{Q}$

$\overline{\Pi x \in A.B[x]}$ = the cartesian product of the family $\bar{B}$ over $\bar{A}$

$\overline{\Pi x \in A.P[x]}$ = the proposition that every $\bar{A}$ is $\bar{P}$

$\overline{A \to B}$ = the functions from $\bar{A}$ to $\bar{B}$

$\bar{N}$ = the natural numbers

$\overline{\Sigma x \in A.B[x]}$ = the disjoint union of the family $\bar{B}$ over $\bar{A}$

$\overline{\Sigma x \in A.P[x]}$ = the proposition that some $\bar{A}$ is $\bar{P}$ or the type of things in $\bar{A}$ that $\bar{P}$ is true of.

$\overline{P \times Q}$ $\bar{P}$ & $\bar{Q}$

$\overline{a =_A b}$ = the proposition that $\bar{a}$ and $\bar{b}$ are the same object of type $\bar{A}$

$\overline{V_n}$ = the type of types or propositions built up from lower types of types

Consequently, we shall use '&' in place of '×' where the intention is propositional. And '∀', '∃', in place of 'Π', 'Σ'.

The intention of the rules is then clear; x∈A corresponds to taking an arbitrary object of type $\bar{A}$;y∈P to an assumption of $\bar{P}$; the = rules are generalizations of the Peano Axioms and the others are generalizations of the IPC rules e.g.

Π-introduction corresponds to

$$\frac{\begin{array}{c}\bar{A}\\ \vdots \\ \bar{B}\end{array}}{\bar{A} \supset \bar{B}} \qquad \text{or} \qquad \frac{\begin{array}{c}\vdots \\ \bar{B}[x]\end{array}}{\forall x.\bar{B}[x]}$$

Π-elimination corresponds to

$$\frac{\bar{A} \supset \bar{B} \qquad \bar{A}}{\bar{B}} \qquad \text{or} \qquad \frac{\forall x.\bar{B}[x]}{\bar{B}[y]}$$

Π-conversion corresponds to

$$\dfrac{\dfrac{\begin{array}{c}\bar{A}\\ \vdots\\ \bar{B}\end{array}}{\bar{A}\supset\bar{B}} \quad \begin{array}{c}\vdots\\ \bar{A}\end{array}}{\bar{B}} \geq \dfrac{\begin{array}{c}\vdots\\ \bar{A}\end{array}}{\bar{B}} \quad \text{or} \quad \dfrac{\begin{array}{c}\vdots\\ \bar{B}[x]\end{array}}{\bar{B}[x]} \geq \dfrac{\begin{array}{c}\vdots\\ \bar{B}[y]\end{array}}{\forall y.\bar{B}[y]}$$

§4. Some theorems of M-L

The real proof theoretic strength of M-L lies in the rules of Σ-elimination and =-elimination. The first allows us to prove a version of the axiom of choice for all types, and the second to prove Leibniz Law and more axioms of identity.

1.4.1. By the axiom of choice at types A,B I mean:

$$a.c.(A,B) =_{df} \Pi x\in A.\Sigma y\in B.R(x,y) \to \Sigma f\in A\to B.\Pi x\in A.R(x,f(x))$$

We also have the stronger version:

$$a.c.(\Pi x\in A.B[x]) =_{df} \Pi x\in A.\Sigma y\in B[x].R(x,y) \to \Sigma f\in\Pi x\in A.B[x].\Pi x\in A.R(x,f(x))$$

For an M-L proof of these see Martin-Löf [1974], page 36.

1.4.2. We can state an axiom of dependent choices for each type:

$$d.c.(A) =_{df} \Pi x\in A.\Sigma y\in A.R(x,y) \to \Pi x\in A.\Sigma F\in N\to A.F(0) = x. \;\&\; \Pi n\in N.R(F(n),F(n+1))$$

To prove this; use the axiom of choice on the antecedent to give an $f\in A \to A$, then define the $F\in N \to A$ by

$$F(0) =_{df} x$$
$$F(s(n)) =_{df} f(F(n))$$

Classically this is an enormously strong principle. Kreisel & Howard [1966], page 327, show that classically $d.c.(N \to N)$ is equivalent to the axiom of monotone bar induction.

1.4.3. And by Leibniz Law we mean the schema:

$$\frac{a=b \qquad C[a]}{C[b]}$$

which is proved by:

$$\frac{\dfrac{z \in a=_A b \qquad \begin{array}{c} x \in A \\ \vdots \\ \lambda y.y \in C[x] \to C[x] \end{array}}{id[z] \in C[a] \to C[b]} \qquad t \in C[a]}{id[z](t) \in C[b]}$$

1.4.4. There is the rule that states that identity is stronger than convertability:

$$\frac{a \text{ conv } b}{a = b}$$

the proof goes:

$$\frac{r(a) \in a=a \qquad \dfrac{a \text{ conv } b}{a = a \text{ conv } a = b}}{r(a) \in a=b}$$

So for example if $p \in \Sigma x \in A.B[x] \to A$ and $q \in \Pi z \in (\Sigma x \in A.B[x]).B[p(z)]$ are the projection functions obtained from the trivial application of Σ-elimination, then we have:

$$\langle p(\langle a,b \rangle), q(\langle a,b \rangle) \rangle = \langle a,b \rangle$$

1.4.5. The converse of Leibniz Law is the principle of identity of indiscernables:

$$id.dis(A) = \Pi f,g \in A.(\Pi \Phi \in A \to V_o.\Phi(f) = \Phi(g). \to f=g)$$

To prove this fix an $f \in A$ and set $\Phi_f(h) =_{df} h=f$

1.4.6. A most useful principle is that every identity statement has a unique proof:

$$\Pi x,y \in a=b.\ x=y$$

which is proved as follows:

$$\frac{x \in A \qquad r(x) \in x =_A x \qquad z \in a=b \qquad r(r(x)) \in r(x)=r(x)}{id[z] \in r(a)=z}$$

§5. Some useful definitions

We can use the principle of 6) above to prove as derived rules the other rules which Martin-Löf gives as part of his definition of $\in$ and conv. They are rules for types with exactly n elements for each n, N_n: and the disjoint union of two types A+B, which has the intended interpretation of the proposition $\bar{A}$ or $\bar{B}$ when A and B are intended as propositions. We state these rules in full now as it will be necessary to take them as primitive later.

Group G_n

N_n-reflection

$$N_n \in V_o$$

N_n-introduction

$$1^*_n,\ldots,n^*_n \in N_n$$

N_n-elimination

$$\frac{z \in N_n \qquad a_1 \in C[1^*],\ldots,a_n \in C[n^*]}{r_n[z] \in C[z]}$$

N_n-conversion

$$\begin{array}{l} r_n(1^*) \text{ conv } a_1 \\ \vdots \\ r_n(n^*) \text{ conv } a_n \end{array}$$

Group F

+-reflection

$$\frac{A \in V_m \qquad B \in V_n}{A+B \in V_{max(m,n)}}$$

+-introduction

$$\frac{x \in A}{i(x) \in A+B} \qquad \frac{y \in B}{j(y) \in A+B}$$

+-elimination

$$\frac{z \in A+B \qquad \begin{array}{c} x \in A \\ \vdots \\ e[x] \in C[i(x)] \end{array} \qquad \begin{array}{c} y \in B \\ \vdots \\ f[y] \in C[j(y)] \end{array}}{dis_{e,f}[z] \in C[z]}$$

+-conversion

$$dis_{e,f}(i(x)) \text{ conv } e[x] \qquad dis_{e,f}(j(y)) \text{ conv } f[y]$$

To obtain the group G derivations define:

$$N_1 =_{df} 1 =_N 1 \; ; \quad 1_1^* =_{df} r(1)$$

and the elimination rule follows by the uniqueness of the object in $1 =_N 1$. And define:

$$N_o =_{df} 0 =_N 1$$

Then the elimination rule:

$$\frac{z \in N_o}{r_o(z) \in C}$$

follows by taking $rec(0) =_{df} N_1$ and $rec(s(n)) =_{df} C$

Using the suggestive notation $T =_{df} N_1$ and $\bot =_{df} N_o$ define the ordering $<$ on N by double recursion:

$$0 < s(n) =_{df} T \qquad \epsilon V_o$$

$$m < 0 \quad =_{df} \perp \qquad \epsilon V_o$$

$$s(m) < s(n) =_{df} m < n \qquad \epsilon V_o$$

then we can define:

$$N_n =_{df} \Sigma x \epsilon N.x < n$$

and i^*_n = (i-1, proof that i-1 < n)

Finally, for the group F rules, define:

$$A+B =_{df} \Sigma x \epsilon N_2.f(x)$$

where: $f(1^*) =_{df} A \times V_{n-1} \epsilon V_{max\ (m,n)}$

$$f(2^*) =_{df} B \times V_{m-1} \epsilon V_{max\ (m,n)}$$

and $i(y) =_{df} (1^*,(y,V_{n-2}))$ and $j(z) =_{df} (2^*,(z,V_{m-2}))$

N.B. I shall use the abbreviations; $\sim A$ for $A \to \perp$ and $a \neq_A b$ for $\sim(a =_A b)$

1.5.2. The definition of N_n given above makes n a real variable appearing in the typesymbol. This is not so in Martin-Löf's usual formalization, in the same way as it is not so for V_n. We can make use of this variable to define for any type A a type $\tilde{A}$ of finite sequences of objects in A:

$$\tilde{A} =_{df} \Sigma n \epsilon N.N_n \to A.$$

In the following chapters I shall make use of standard notation for sequences and their codes, but it is always understood that the formal counterpart of these objects can be defined in $\tilde{A}$.

1.5.3. Also for any type A, a <u>species of objects of type A</u> is an object of type

$$A \to V_k$$

for some k. More specifically I shall refer to such an object as a kth. order species of objects of type A.

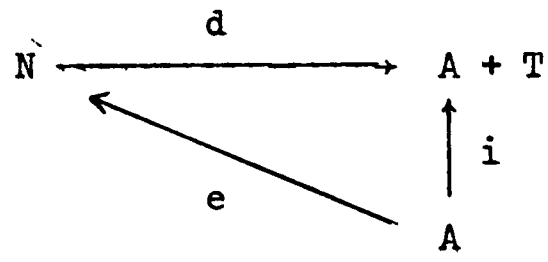
Then we say that for the species $F \in A \to V_k$, that an object $a \in A$ is a member of the species if F(a). A species F of members of A determines a 'subtype' of A, namely:

$$\Sigma x \in A.F(x)$$

and it is sometimes easier to think of the species as being that type.

As a notational convenience we shall use the abbreviations $\Pi x \in F.B[x]$ for $\Pi x \in A.F(x) \to B[x]$, and $\Sigma x \in F.B[x]$ for $\Sigma x \in A .F(x) \ \& \ B[x]$.

1.5.4. A type A is countable, informally, if there is a function which 'encodes' it into the natural numbers and one which 'decodes' it in such a way that the code of an object is decoded back to itself. But we must allow an empty type to be countable, and in that case, there can be no function from the natural numbers to that type; no decoding of that type. To allow for this, the decoder function must take a number not to the type in question but to the disjoint union of this type with a singleton type. So a type is countable iff the following diagram commutes.



More formally define the predicate Count( ) as follows:

$$\text{Count}(A) =_{df} \Sigma f \in A \to N.\Sigma g \in N \to A+T.\Pi x \in A.g_\circ f(x) = i(x)$$

And it may be more convenient, when we know that the type in question in non-empty, to use the notion of strict countability:

$$\text{S.count}(A) =_{df} \Sigma f \in A \to N.\Sigma g \in N \to A.\Pi x \in A.g_\circ f(x) = x$$

Finally, a species F is said to be countable or strictly countable if $\Sigma x \in A.F(x)$ is.

Footnote 1

It suffices to give the definition of substitution for one variable.

If for $x \in A$, $c[x]$ is the variable x then when $a \in A$ $c[a]$ is a

If $c[x]$ is $f(f;x)$ then $c[a]$ is $(f;a)$

Footnote 2

Strictly, the definition of 'depends' should be given simultaneous with the clauses of group A:

If the variable x has typesymbol A and A depends on the variables $x_1, x_2, \ldots, x_m$ then x depends on $x_1, x_2, \ldots, x_m$.

If the n-ary function constant f is introduced as on page 8, and a_1 depends on the variables $x_{1_1}, x_{1_2}, \ldots, x_{1_{k_1}}$ and a_2 depends on $x_{2_1}, x_{2_2}, \ldots, x_{2_{k_2}}$ etc. then

$(f;a_1,a_2,\ldots,a_n)$ depends on $x_{1_1}, x_{1_2}, \ldots, x_{2_1}, x_{2_2}, \ldots, x_{n_{k_n}}$.

If for some i, a_i is a variable x_i then

$(f;a_1,a_2,\ldots,a_n)$ depends on $x_{1_1}, \ldots, x_{i_{k_i}}, x_i, x_{(i+1)_1}, \ldots, x_{n_{k_n}}$.

Chapter 2

MODELS OF M-L

In his paper [1972], Martin-Löf formulates a notion of model which is general enough to embrace versions of every kind of 'models' of intuitionistic systems hitherto constructed. And whereas these models may have been constructed in either a classical or an intuitionistic framework, his notion was purely constructive. In fact, it was formulated inside M-L.

Proceeding, then in the spirit of this paper we extend the definition given there to that of a model of M-L itself. Such a model will not be constructable completely inside M-L, naturally, but it will be seen that each stage of the construction is. Roughly, we define objects of M-L over the typestructure of M-L and prove relationships between these objects inside M-L. However, the definition of these objects does not take place inside M-L.

The idea is that the typesymbols and terms are interpreted as objects of M-L, and then we proceed to show that the interpretation of $\in$ and conv is standard.

The natural approach would be to make models by interpreting typesymbols as typesymbols and terms as terms; and then proving that for a term a of type A, the term $\bar{a}$ which interprets a is a term of the typesymbol $\bar{A}$ which interprets A. This is not directly possible, however, for every typesymbol is a term, of some universe, itself. This means that a typesymbol could more or less only be interpreted as itself. To overcome this we must interpret a typesymbol qua term as a term and qua typesymbol as a typesymbol. This is achieved by having a function which takes the interpretation of a typesymbol qua term - a term - to the interpretation of the typesymbol qua typesymbol - a typesymbol. The principal model forming equipment, then, is a collection of types which are to be the interpretations of the universe typesymbols qua typesymbols, and a type

valued function for each of these types, which is to be the function discussed above.

The syntax of M-L makes it necessary to interpret not just terms and typesymbols, but all formal expressions. And separately to prove that the terms of a typesymbol were interpreted as terms of the typesymbol which interpreted that typesymbol qua typesymbol.

To summarize we can make the definition:

1. The formal definition

A model of M-L comprises the following:

1) A sequence of types $Typ_0, Typ_1, Typ_2, \ldots\ldots$

2) A sequence of functions $Obj_0, Obj_1, Obj_2, \ldots\ldots$

such that $Obj_i \in Typ_i \to V_i$

3) A function $|.|$, which takes formal expressions to formal expressions, defined over the definition of formal expression; such that

$$Obj_{i+1}(|V_i|) = Typ_i$$

4) Simultaneously

i) an assignment for each variable $\xi \in Obj(|A|)$ of a variable $\underline{\xi} \in A$.

ii) a proof over the definition of $\in$ and conv that

a) if $a \in A$ is a closed term then $|a| \in Obj(|A|)$

b) if a,b are closed terms and a conv b then

$$|a| \text{ conv } |b|$$

c) if f is a function constant with index/value typesymbols:

$$A_1, A_2[x], \ldots, A_n[x_1, \ldots, x_{n-1}]/A[x_1, \ldots, x_n]$$

then we have for

$$\xi_1 \in Obj(|A_1|), \xi_2 \in Obj(|A_2[\underline{\xi_1}]|), \ldots, \xi_n \in Obj(A_n[\underline{\xi_1}, \ldots, \underline{\xi_{n-1}}]|)$$

a term:

$$|(f;\underline{\xi_1}, \ldots, \underline{\xi_n})| \in Obj(|A[\underline{\xi_1}, \ldots, \underline{\xi_n}]|)$$

where $|(f;\underline{\xi}_1,\ldots,\underline{\xi}_n)|$ is a term $b[\xi_1,\ldots,\xi_n]$. By the rule of $\prod$-introduction we can then introduce a function

$\bar{f} \in \prod\xi_1 \in Obj(|A_1|).\prod\xi_2 \in Obj(|A_2[\xi_1]|).\ldots\ldots\ldots Obj(|A_n[\xi_1,\ldots,\xi_n]|)$

such that: $(ap;\bar{f},\xi_1,\ldots,\xi_n) = |(f;\underline{\xi}_1,\ldots,\underline{\xi}_n)|$

and we require that:

$$(ap;\bar{f},|a_1|,\ldots,|a_n|) = |(f;a_1,\ldots,a_n)|.$$

Writing $|(f;\underline{|a_1|},\ldots,\underline{|a_n|})|$ as the result of substituting $|a_i|$ for ξ_i in $|(f;\underline{\xi}_1,\ldots,\underline{\xi}_n)|$, the requirement is that:

$$|(f;\underline{|a_1|},\ldots,\underline{|a_n|})| = |(f;a_1,\ldots,a_n)|$$

N.B. Subscripts on Obj's are systematically dropped.

Taking 4)i) and 4)ii)c) together we can show how the problematic rule of convertability is modelled;

$$|a_1| \text{ conv } |b_1|,\ |a_2| \text{ conv } |b_2|,\ldots,|a_n| \text{ conv } |b_n|$$

then: $$|(f;a_1,\ldots,a_n)| = (ap;\bar{f},|a_1|,\ldots,|a_n|)$$

$$\text{conv } (ap;\bar{f},|b_1|,\ldots,|b_n|) = |(f;b_1,\ldots,b_n)|$$

§2. The Model of Normal Terms

In order to exemplify this definition we shall use the familiar model of normal terms, as worked out for M-L in Martin-Löf [1974], for example. In this model, a term is to be interpreted as a pair. The first coordinate being a closed normal term that it reduces to, and the second a proof that that term is 'computable', the computability predicate com( ) being defined simultaneously with the interpretation. Similarly a typesymbol is interpreted as a pair, the first coordinate of which is the normal term to which it reduces, as before, but the second coordinate

is the predicate for that typesymbol. Using the framework as described above we proceed as follows:

i) $Typ_i =_{df} \Sigma X \in V_i . \Sigma x \in X.cn(x) \to V_i$. where $\Sigma x \in A.cn(x)$ is the type of closed normal terms of type A.

ii) For $\Phi \in Typ_i$

$$Obj(\Phi) =_{df} \Sigma x \in p(\Phi).q(\Phi)(x)$$

then we can define

iii) $|V_i| =_{df} (V_i, \lambda X. \Sigma x \in X.cn(x) \to V_i.)$

iv) $|A| =_{df} (A', com_{A'})$

v) $|a| =_{df} (a',$ proof that $com_{A'}(a'))$

and show that $a' \in \Sigma x \in A'.cn(x)$ and $com_{A'}(a')$

§3. Realizability Models

The idea behind realizability is that a statement considered intuitionistically requires some 'completing of information'. For example the statement that $\forall x.A(x)$ requires a function which takes a natural number x to something that completes the information that A(x).

So, in a sense, M-L requires no completing of information. For each provable typesymbol of M-L, we can provide a constructive proof; the information given by the typesymbol can be completed by its proof. However, information can be completed in many ways and that is the value of realizability models. The straightforward way of completing information provides the smallest model of M-L; the term model, where we complete information by a standard proof. Whereas with realizability, a proof a of A will be interpreted as an object, $\bar{a}$, which realizes A, we may be able to realize typesymbols which do not have proofs. Provided that we are careful to define what it is to realize a typesymbol (over the type structure), in such a way that no disprovable statements are realized, this will lead to consistency results.

The models we construct here are ipso-facto models of HA and other systems of intuitionistic mathematics; but it cannot be assumed that, for example, the number realizability model which I shall construct will realize the M-L equivalents of all the statements realized by (say) Kleene [1952] in his number realizability model. Many usual practices cannot be adopted by us, if we adhere to the definition of model given above. One such practice is to define 'x realizes $A \to B$' by cases, according as $A,B=N$ or not.

Troelstra argued [1971] page 3, that it would be unprofitable to give in detail a general scheme for either axiomatizing existing notions of realizability for intuitionistic analysis or prescribing axiomatically a new notion. However, in the case of M-L, the complexity of the type structure compels us to do just that. We need to set up a general framework.

We start off by considering a non-empty domain of realizing objects and without reference to the model forming apparatus (Typ, Obj etc) we can specify certain properties of this domain which are essential to the subsequent model. We must be able to pair and unpair members of the domain, so that an object that realizes $\Sigma x \in A.B[x]$ is composed of one that realizes $B[a']$, where a' in A, is uniquely determined by the object that realizes A.

The natural numbers will have to be realized by a countable species of members of the domain, and in order to realize a proposition $B[x]$ defined by induction on the natural numbers we require a 'variable' realizing object, dependent on each object of the species, which in turn realizes $B[0]$ and $B[s(y)]$. All this is relatively simple; it is in realizing $\Pi x \in A.B[x]$ that more is required than the domain of realizing objects. Given for each $x \in A$, an object which realizes $B[x]$, we must find an object to realize $\Pi x \in A.B[x]$. And from an object that realizes $\Pi x \in A.B[x]$ we should be able to realize $B[x]$ for each $x \in A$. This suggests that the objects which realize $B[x]$ should be given functionally and that the object that realizes $\Pi x \in A.B[x]$ codes this function. From such a coded function we could retrieve

the object that realizes B[x] for each x∈A. To this end we need a two place operation on the domain, which is conditional on the first being the code of a function, defined at the second. So, along with the domain of realizing objects, we require a species of functions on the domain which can be encoded into objects of the domain.

In more detail, we define a realizing structure (r.s.) to be:

1) A non-empty type D, with distinguished member $\mathbb{0}$, st. $D \in V_0$ or V_1

2) an equivalence relation $\simeq$ on D

3) the following operations on D:

i) [,] - pairing

ii) $\mathbb{p}$ and $\mathbb{q}$, so that $\mathbb{p}([\alpha,\beta]) \simeq \alpha$

and $\mathbb{q}([\alpha,\beta]) \simeq \beta$

iii) $ - successor, so that: $\forall n.\$^{n+1}(0) \neq \$^{n}(0)$

where $\$^n$ is defined by:

$$\$^0 =_{df} \lambda\alpha.\alpha$$

$$\$^n =_{df} \$.\$^{n-1}$$

And we could write $\$^1(0)$ as $\mathbb{1}$, $\$^2(0)$ as $\mathbb{2}$ etc.

iv) $(\supset,)$ - conditional, so that

$$(\mathbb{0} \supset \alpha,\beta) \simeq \alpha$$

$$(\$(\gamma) \supset \alpha,\beta) \simeq \beta$$

v) |-application, which is a partial operation, in that it is only defined under the condition C.

i.e. $| \in \sum x \in D \times D.C(p(x),q(x)). \longrightarrow D$; and we write

$|((\alpha,\beta), \text{proof that } C(\alpha,\beta)))$ as $\alpha|\beta$.

4) for each sequence of species on D ; $D_1, D_2, \ldots\ldots, D_n$ with :

$$D_j \in D \longrightarrow V_{i_j}$$

a species:

$$S_{D_1,\ldots,D_n} \in (D_1 \times \ldots\ldots \times D_n \longrightarrow D) \longrightarrow V_\alpha$$

with the following properties:

i) $\$ \in S_D$

ii) closure under the operations

a) $\forall f,g,h \in S_{D_1,\ldots D_n}$

$$\lambda\alpha_1\ldots\alpha_n.[f(\alpha_1,\ldots,\alpha_n),g(\alpha_1,\ldots,\alpha_n)] \in S_{D_1\ldots D_n}$$

$$\lambda\alpha_1\ldots\alpha_n.(f(\alpha_1,\ldots,\alpha_n) \supset g(\alpha_1,\ldots,\alpha_n),h(\alpha_1,\ldots,\alpha_n)) \in S_{D_1\ldots D_n}$$

b) $\forall f,g \in S_{D_1\ldots D_n}.(\forall\alpha_1\in D_1,\ldots\ \forall\alpha_n\in D_n.$

$C(f(\alpha_1,\ldots,\alpha_n),g(\alpha_1,\ldots,\alpha_n))$

$$\lambda\alpha_1,\ldots,\alpha_n.\ f(\alpha_1,\ldots,\alpha_n)\,|\,g(\alpha_1,\ldots,\alpha_n) \in S_{D_1\ldots D_n}$$

iii) $\forall f,g \in S_{D_1\ldots D_n}.\ \forall\alpha_1,\beta_1\in D_1,\ldots\ \forall\alpha_n,\beta_n \in D_n$

$$\alpha_1\simeq\beta_1\ \&\ \alpha_2\simeq\beta_2\ \&\ \ldots\ \&\ \alpha_n\simeq\beta_n$$

$$\Rightarrow\quad f(\alpha_1,\ldots,\alpha_n)\simeq g(\alpha_1,\ldots,\alpha_n)$$

iv) the important application condition

$$\forall f\in S_{D_1\ldots D_{n+1}}.\ \exists g\in S_{D_1\ldots D_n}.\ \forall\alpha_1\in D_1,\ldots\forall\alpha_n\in D_n.\ \forall\alpha\in D_{n+1}$$

$C(g(\alpha_1,\ldots,\alpha_n),\alpha)\ \&\ g(\alpha,\ldots,\alpha_n)\,|\,\alpha \simeq f(\alpha_1,\ldots,\alpha_n)$

and we write $\Lambda_\alpha f[\alpha]$ for g .

v) $\lambda x_1,\ldots x_n.x_i \in S_{D_1\ldots D_n}$

1) and 3)iii) give us a countable species $\mathbb{N}$ of members or D which we can define as follows:

$$\mathbb{N}(\alpha) =_{df} \quad \Sigma n\in N.\alpha = \$^n(0)$$

In fact, $\mathbb{N}$ is strictly countable, as we have a definite member; 0 together with the proof that $\mathbb{N}(0)$. The coding and decoding are achieved as follows:

$$e((0, \text{proof that } \mathbb{N}(0))) =_{df} 0$$

$$e((\$(\alpha), \text{proof that } \mathbb{N}(\$(\alpha)))) =_{df} s(e((\alpha, \text{proof that } \mathbb{N}(\alpha)))$$

$$d(0) =_{df} (0, \text{proof that } \mathbb{N}(0))$$

$$d(s(n)) =_{df} (\$(d(n)), \text{proof that } \mathbb{N}(\$(d(n))))$$

And recursion is then made possible by defining a fixed point operator Fx in the usual way, with $Fx \in S_D$:

$$Fx(f) =_{df} \Lambda x.f|(x|x)|\Lambda x.f|(x|x)$$

4)iv) is just (for n places) the requirement of a coded function.

Now, given a r.s. we can construct a model (said to be the model over the r.s.) by interpreting a typesymbol as a pair, the first coordinate of which is a realizing object (it realizes the universe typesymbol of the typesymbol) and the second, the species of objects that realize it. Then a term of that typesymbol is interpreted as a pair, the first coordinate of which is a realizing object and the second a proof that this object is in the species that realizes the typesymbol. Making use of the apparatus of the definition of model

i) $Typ_k =_{df} D \times (D \to V_k)$

ii) for each $\Phi \in Typ_k$

$$Obj_k(\Phi) =_{df} \Sigma x \in D.q(\Phi)(x)$$

The application property of the r.s. is used by defining simultaneous with the definition of $|.|$,

So, when

defining $|.|$, if for

$$x_1 \in A_1, x_2 \in A_2[x_1], \ldots, x_n \in A_n[x_1, \ldots, x_{n-1}]$$

we have

$$b[x_1, \ldots, x_n] \in A[x_1, \ldots, x_n]$$

then the induction hypothesis requires that for

$$\xi_1 \in Obj(|A_1|),\ \xi_2 \in Obj(|A_2[\xi_1]|), \ldots,\ \xi_n \in Obj(|A_n[\xi_1, \ldots, \xi_{n-1}]|)$$

we define

$$|b[\xi_1, \ldots, \xi_n]| \in Obj(|A[\xi_1, \ldots, \xi_n]|)$$

And in addition

$$\|b[\underline{\xi}_1,\ldots,\underline{\xi}_n]\| \in {}^{S}q(|A_1|),\ldots\ldots,q(|A_n[\xi_1,\ldots,\xi_{n-1}]|)$$

This takes place throughout the definition of $|.|$ and so provides, by the application property, a realizing object for $\lambda x b[y_1,\ldots,y_{n-1},x]$ i.e.

$$p(|\lambda x.b[\underline{\xi}_1,\ldots,\underline{\xi}_{n-1},x]|) =_{df} \lambda\alpha.\ \|b[\xi_1,\ldots,\xi_n]\|(p(\xi_1),\ldots,p(\xi_n),\alpha)$$

To show that such a procedure does produce a model of M-L according to the definition we must give the definitions of $|.|$, $\|.\|$ in full and check the clause 4)ii). But we delay this until the end of the chapter, in order to exemplify now the definition of an r.s.

§4. Some examples of realizability models.

In each case we need only give the underlying r.s.

2.4.1. Number realizability. The version for HA and IPC appears e.g. in Kleene [1952], and we adopt notations found there.

1) $D =_{df} N$ and has distinguished element $\mathbb{0} =_{df} 0$

2) $n \simeq m =_{df} n=m$

3) i) $[n,m] =_{df} 2^n 3^m$

ii) $\mathbb{p}(r) =_{df} (r)_0$, $\mathbb{q}(r) =_{df} (r)_1$

iii) $\$(n) =_{df} s(n)$

iv) $(n \supset m,r) =_{df} rec(n)$ where

$$rec(0) =_{df} m$$

$$rec(s(t)) =_{df} r$$

v) $n|m =_{df} \{n\}(m)$ and

$$C(n,m) =_{df} \Sigma w\in N.T(n,m,w) \ \&\ \Pi v\in N.(v<w \rightarrow \sim T(n,m,v))$$

4) $S_{D_1,\ldots,D_n}(f) =_{4f} \Sigma e\in N.\forall\alpha_1\in D_1,\ldots\ldots\forall\alpha_n\in D_n.\Sigma w\in N.$

$$T(e,\alpha_1,\ldots,\alpha_n,w)\ \&\ \Pi\, v<w\ .\sim T(e,\alpha_1,\ldots,\alpha_n,v)$$

$$\&\ f(\alpha_1,\ldots,\alpha_n) = U(w$$

i),ii)a) are trivial & ii)b) follows from the fact that $U(\mu y.T(n,m,y))$ is recursive and total if $C(n,m)$.

iii) is just Leibniz Law

iv) is just the $S^{i,m}$ theorem, so

$$\Lambda\alpha\phi[\alpha] =_{df} S^{i,m}(gn(\phi),\beta_1,\ldots,\beta_n)$$

N.B. from 3)iii) and 1), $\mathbb{N}$ is the total species of natural numbers.

Properties of number realizability

1) $\Pi\eta,\mu \in Obj(|N|).Obj(|\underline{n}=\underline{\mu}|) \leftrightarrow p(\eta) = p(\mu) \leftrightarrow \underline{n} = \underline{\mu}$

This is actually true of every realizability model, but is so obvious here, where the natural numbers are realized by themselves.

2) Every primitive recursive function ϕ is such that $p(|\phi(\underline{n})|) = \phi(p(\eta))$

i) ϕ is the zero function, $\lambda x.0$

$$p(|\lambda x.0(n)|) = p(|0|) = 0 = \lambda x.0(p(|n|))$$

ii) ϕ is the successor function, s

$$p(|s(n)|) =_{df} s(p(|n|))$$

iii) ϕ is a projection function, U_i^m

$$p(|U_i^m(n_1,\ldots,n_m)|) = p(|n_i|)$$
$$= U_i^m(p(|n_1|),\ldots,p(|n_m|))$$

iv) ϕ is obtained by substitution

Assume $p(|\psi(y_1,\ldots,y_m)|) = \psi(p(|y_1|),\ldots,p(|y_m|))$

$$p(|\chi_i(z_1,\ldots,z_r)|) = \chi_i(p(|z_1|),\ldots,p(|z_r|))$$

$$p(|\psi(\chi_1(n_1,\ldots,n_r),\chi_2(n_1,\ldots,n_r),\ldots,\chi_m(n_1,\ldots,n_r))|)$$

$$= \psi(p(|\chi_1(n_1,\ldots,n_r)|),\ldots,p(|\chi_m(n_1,\ldots,n_r)|))$$

$$= \psi(\chi_1(p(|n_1|),\ldots,p(|n_r|)),\ldots,\chi_m(p(|n_1|),\ldots,p(|n_r|)))$$

v) ϕ is obtained by recursion

Assume $p(|\psi(n_1,\ldots,n_m)|) = \psi(p(|n_1|),\ldots,p(|n_m|))$

$$p(|\chi(n_1,\ldots,n_m,n,t)|) = \chi(p(|n_1|),\ldots,p(|n_m|),p(|n|),p(|t|))$$

$$p(|rec(n_1,\ldots,n_m n)|) =$$

$$F\chi(\Lambda g\Lambda z.z \supset \psi(p(|n_1|),\ldots,p(|n_m|)),\chi(p(|n_1|),\ldots,p(|n_m|),$$

$$pd(z),g(pd(z)))|p(|n|)$$

$$= rec(p(|n_1|),\ldots,p(|n_m|),p(|n|))$$

3) Every primitive recursive predicate P is such that;

$$Obj(|P(\underline{\xi})|) \leftrightarrow P(p(\xi))$$

This follows from 1) as

$$P(n) \leftrightarrow \phi_p(n)=0$$

$$Obj(|P(n)|) \leftrightarrow Obj(|\phi_p(n)=0|)$$

$$\leftrightarrow p(|\phi_p(n)|)=p(|0|)$$

$$\leftrightarrow \phi_p(p(|n|))=0$$

$$\leftrightarrow P(p(|n|))=0$$

2.4.2. Function realizability

The original formalization for less expressive systems can be found in Kleene-Vesley [1965] and Troelstra [1971(b)]. But the model here differs essentially from theirs even for the finite types e.g. a number theoretic function is realized by itself in the usual formulation and we cannot give special treatment to any type. However, we adopt their notation.

1) $D =_{df} N \to N$ and $\mathbb{0} =_{df} \lambda x.x$

2) $f \simeq g =_{df} \Pi x \in \ldots f(x)=g(x)$

3) i) $[f,g] = \lambda x.2^{f(x)}3^{g(x)}$

ii) $p =_{df} \lambda f.\lambda x.(f(x))_0$; $q =_{df} \lambda f.\lambda x.(f(x))_1$

iii) $\$(f) =_{df} \lambda x.f(x) + 1$

iv) $(f \supset g,h) =_{df} rec(f(0))$ where $rec(o) =_{df} g$ and $rec(s(n)) =_{df} h$

v) $f|g =_{df} \lambda t.f(t*\bar{g}(n_t)) \div 1$

where $n_t =_{df} \mu y.f(t*\bar{g}(y)) \neq 0$ and

$C(f,g) =_{df} \Pi t\in N.\Sigma y\in N.f(t*\bar{g}(y)) \neq 0$

4) $S_{\sigma_1,\ldots,\sigma_n}(f) =_{df} \Sigma e\in N.\ \Pi\alpha_1\in\sigma_1 \ldots\ldots \Pi\alpha_n\in\sigma_n.\ \Pi t\in N.\Sigma w\in N$

$T^{1,0}_n(e,\alpha_1(w),\ldots,\alpha_n(w),w)\ \&\ \Pi v<w.\sim T^{1,0}_n(e,\alpha_1(v),\ldots,\alpha_n(v),v)$

$\&\ f(\alpha_1,\ldots,\alpha_n,\tau) = U(w)$

where $T^{1,0}_n$ and U are the usual predicate and function of Kleene.

i.e. f is totally recursive on its domain.

i),ii)a) the S_i are clearly closed under the operations.

ii)b) $\lambda t.f(t*\bar{g}(\mu y.f(t*\bar{g}(y))\neq 0))$ is recursive in f,g

and if C(f,g) then it is total.

iii) is a property of recursive functions.

iv) is Kleene's lemma 8.1, page 90 in [1965]

N.B. N is the species of constant functions.

2.4.3. Set realizability. This is not to be found in the literature but is based on Scott's models of the lambda calculus, in order to use their excellent continuity properties. I shall use notions from and refer to proofs in his paper [1973].

1) $D =_{df}$ the type of effectively enumerable sets of natural numbers;

$N \to N+T\ .\ \Phi =_{df} \lambda x.i(0)$

Whilst I shall use set theoretic symbols informally, they can be defined

$x\varepsilon X =_{df} \Sigma m\in N.X(m) = i(x)$

$X \subseteq Y =_{df} \Pi x\in N.\ x\varepsilon X \to x\varepsilon Y\ .$

$X \equiv Y =_{df} X\subseteq Y\ \&\ Y\subseteq X$

and if we have proved :

$\Pi x\in D.\ x\varepsilon A \leftrightarrow \Phi(x)$

then we shall write:

$X = \{x|\ \Phi(x)\}$

2) $X \simeq Y \;=_{df}\; X \equiv Y$

3) i) firstly define $\{0\} \times X$ by:

$$\{0\}\times X \;=_{df}\; \lambda n.\begin{cases} i((0,n_0)) & \text{if } i(n_0)=X(n_1) \\ j(1^*) & \text{otherwise} \end{cases}$$

and similarly $\{1\}\times Y$. Then:

$$[X,Y] \;=_{df}\; \lambda m.\begin{cases} \{0\}\times X\ (m_0) & \text{if } i(m_0)=\{0\}\times X\ (m_1) \\ \{1\}\times Y\ (m_0) & \text{if } i(m_0)=\{1\}\times Y\ (m_1) \\ j(1^*) & \text{otherwise} \end{cases}$$

And informally, $X,Y = \{x \mid x_0=0 \ \& \ x_1 \in X \text{ or } x_0=1 \ \& \ x_1 \in Y\}$

ii) the projections are defined by:

$$p(Z) = \lambda x.\begin{cases} i(x_0) & \text{if } i(0,x_0)=Z(x_1) \\ j(1^*) & \text{otherwise} \end{cases}$$

and similarly for q .

iii) $$\$(X) =_{df} \lambda x.\begin{cases} i(x+1) & \text{if } X(x) \\ j(1^*) & \text{otherwise} \end{cases}$$

And informally, $\$(X) = \{n+1 \mid n \in X\}$.

iv) $(X \supset Y,Z) = \{n \in Y / O \in X\} \cup \{m \in Z \mid \Sigma n \in N.n+1 \in X\}$

which formally is obtained as for $X|Y$.

v) $X|Y =_{df} \{x \mid \Sigma n \in N.e_n \subseteq Y \ \&\ (n,x) \in X\}$

where e_n is the n-th finite member of D given by

$$e_n =_{df} \lambda x.i(k_x)$$

$$n = \sum_{i \leq m} 2^{k_i} \quad \text{and} \quad k_i \leq k_j \text{ for } i \leq j$$

and $(n,m) =_{df} \frac{1}{2}\cdot(n+m)\cdot(n+m+1)+m$

More formally: $\exists n.(e_n \subseteq Y.(n,x) \in X)$ is of the form $\exists m.P(x,m)$ where P is a decidable relation. So put

$$X|Y =_{df} \lambda t.\begin{cases} i((t)_0) & \text{if } P((t)_0,(t)_1) \\ j(1^*) & \text{o.w.} \end{cases}$$

$C(X,Y) =_{df} T$, i.e. the operation is total.

4) $S_{D_1 \cdots D_n}(f) =_{df} f$ is continuous on its domain.

From Scott we have: X is continuous

$\iff e_m \subseteq X(x)$ iff $\Sigma n \in N.e_n \subseteq x \ \&\ e_m \subseteq X(e_n)$

$\iff m \in X(x)$ iff $\Sigma n \in N.e_n \subseteq x \ \&\ m \in X(e_n)$.

and that X is continuous

$\implies$ if $x \subseteq y$ then $X(x) \subseteq X(y)$

i)ii) the reader is referred to Scott [1973]

iii) follows from monotonicity

iv) is fulfilled by setting;

$$\Lambda\alpha.\phi =_{df} \{ (n,m) \mid m \varepsilon \phi(e_n) \}$$

the graph of ϕ; and noting that

$$\Lambda\alpha.\phi|\alpha =_{df} \{k \mid \Sigma\ell \in N.e_\ell \subseteq \alpha \ \& \ k \varepsilon \phi(e_\ell)\}$$

$$= \{k \mid k \varepsilon \phi(\alpha)\} = \phi(\alpha)$$

$$(k,\ell) \varepsilon \Lambda\alpha.\phi(\beta) =_{df} \{ (m,n) \mid n \varepsilon \phi(e_n)(\beta) \}$$

$$\ell \varepsilon (e_k)(\beta) \text{ iff } \Sigma r \in N.e_r \subseteq \beta \ \& \ \ell \varepsilon \phi(e_n)(e_r)$$

$$\text{iff } \Sigma r \in N.e_r \subseteq \beta.\ (k,\ell) \varepsilon \Lambda\alpha.\phi(e_r)$$

i.e. $\Lambda\alpha.\phi$ is continuous

N.B. $\mathbb{N}$ is the species of singleton sets.

§5. The proof of §1

For the sake of notational ease, we consider only one parameter instead of the required n. Also the definition of $|.|$ and $\|.\|$ are given simultaneously with the proofs that, in effect, the interpretation of $\in$ and conv is standard. This again is for ease of reading: and the stages of the definition are given groupwise. In each group we write the interpretations first and then make the proofs. We sometimes use the simplified notation of chapter 1, §2 for the interpretations.

Group A

i) $|\xi| =_{df} \xi$ where ξ is a variable

$\|\xi\| =_{df} p(\xi)$

ii) $|V_n| =_{df} (\mathbb{O}, \lambda\alpha.D \to V_n)$ then $Obj(|V_n|) =_{df} Typ_n$

The proofs are as follows:

$$\frac{\xi \in Obj(|A|)}{|\underline{\xi}| =_{df} \xi \in Obj(|A|)}$$

and

$$|V_n| \in D \times (D \to V_{n+1})$$

$$=_{df} Typ_{n+1} =_{df} Obj(|V_{n+1}|)$$

The others in this group follow straightforwardly from the <u>standard</u> interpretation of conv and in the case of the main rule of convertability, the proof has already been given above.

In what follows, we assume that $\xi \in Obj(|A_1|)$

<u>Group B</u>

i) $p(|\Pi(A[\underline{\xi}],B[\underline{\xi},x])|) =_{df} 0$

$q(|\Pi(A[\underline{\xi}],B[\underline{\xi},x])|)$ is the species of realizing objects given by:

$q(|\Pi(A[\underline{\xi}],B[\underline{\xi},x])|)(\alpha) =_{df} \Pi\zeta \in Obj(|A[\underline{\xi}]|).\ \Sigma\chi \in Obj(|B[\underline{\xi},\underline{\zeta}]|).$

$$\alpha|p(\zeta) \simeq p(\chi) \ \&\ C(\alpha,p(\zeta))$$

ii) $p(|\lambda b[\underline{\xi},x]|) =_{df} \Lambda\alpha.\|b[\underline{\xi},x]\|(p(\xi),\alpha)$

$q(|\lambda b[\underline{\xi},x]|)$ is the proof that $p(|\lambda b[\underline{\xi},x]|)$ is in the species $q(|\Pi(A[\underline{\xi}],B[\underline{\zeta},x])|)$, which is defined as follows:

$q(|\lambda b[\xi,x]|) =_{df} \lambda\zeta.(|b[\underline{\xi},\underline{\zeta}]|,(p_1,p_2))$

where p_1 is the proof that

$$\Lambda\alpha.\|b[\underline{\xi},x]\|(p(\xi),\alpha)|p(\zeta) \simeq p(|b[\underline{\xi},\underline{\zeta}]|)$$

obtained from the induction hypothesis; and p_2 is the proof that

$$C(\Lambda\alpha.\|b[\underline{\xi},x]\|(p(\zeta),\alpha),p(\zeta))$$

obtained from the definition of the r.s.

Also $\|\lambda b[\underline{\xi},x]\|(\beta) =_{df} \Lambda\alpha.\|b[\underline{\xi},x]\|(\beta,\alpha)$

iii) $|(ap;\underline{\xi},f[\underline{\xi}],a[\underline{\xi}])| =_{df} p(q(|f[\underline{\xi}]|)(|a[\underline{\xi}]|)$ and

$\|(ap:\underline{\xi},f[\underline{\xi}],a[\underline{\xi}])\|(\alpha) =_{df}$ illegible

Π-reflection. Assume that $|A[\underline{\xi}]| \in Obj(|V_n|)$ and for $\zeta \in Obj(|A[\underline{\xi}]|)$, $|B[\underline{\xi},\underline{\zeta}]| \in Obj(|V_m|)$. Then $Obj(|A[\underline{\xi}]|) \in V_n$, $Obj(|B[\underline{\xi},\underline{\zeta}]|) \in V_m$, $\simeq$ and $C \in D \to (D \to V_o)$ and so $|\Pi(A[\underline{\xi}], B[\underline{\xi},x])| \in Obj(|V_{max(n,m)}|)$.

Π-introduction. Obvious.

Π-elimination. Assume that $|f[\underline{\xi}]| \in Obj(|\Pi(A[\underline{\xi}], B[\underline{\xi},x])|)$ and $a[\underline{\xi}] \in Obj(|A[\underline{\xi}]|)$, then

$|(ap;\underline{\xi},f[\underline{\xi}],a[\underline{\xi}])| \in Obj(|B[\underline{\xi},|a[\underline{\xi}]|]|)$

$= Obj(|B[\underline{\xi},a[\underline{\xi}]]|)$ by definition.

Π- conversion. $|(ap;a_1,(\lambda b[\underline{\xi},x];a_1),a)| = |b[|a_1|,|a|]| = |b[a_1,a]|$

Group C

i) $|N| =_{df} (\mathbb{0},\mathbb{N})$.

ii) $|0| =_{df} d(0)$

iii) $|(s;\underline{n})| =_{df} d(s(e(n)))$

$\|(s;\underline{n})\| =_{df}$ illegible

iv) $|(rec(a[\underline{\xi}],b[\underline{\xi},x,y]);\underline{\xi},\underline{n})| = (rec(|a[\underline{\xi}]|,|b[\underline{\xi},n,\underline{\zeta}]|);\underline{\xi},e(n))$

$\|(rec(a[\underline{\xi}],b[\underline{\xi},x,y]);\underline{\xi},\underline{n})\|'(\alpha,\beta) =_{df} Fx(\Lambda g.\Lambda z.z\supset\|a[\underline{\xi}]\|(\alpha),\|b[\underline{\xi},\underline{\chi},\underline{\zeta}]\|$

$(\alpha,pd(z),g(pd(z)))\ )|\beta$

where pd is the opposite operation to \$. This follows from 4)i)a).

N-reflection. $\mathbb{N} \in D \to V_o$

N-introduction. Obvious.

N-elimination. Assume that for $\chi \in Obj(|N|)$ & $\zeta \in Obj(|A[\underline{\xi},\underline{\chi}]|)$

$|b[\underline{\xi},\underline{\chi},\underline{\zeta}]| \in Obj(|A[\underline{\xi},(s;\underline{\chi})]|)$; and $|a[\underline{\xi}]| \in Obj(|A[\underline{\xi},0]|)$

$= Obj(|A[\underline{\xi},\underline{d(0)}]|)$

Then for $n \in Obj(|N|)$, $e(n) \in N$ and we have:

$$\frac{n \in N \qquad \zeta \in Obj(|A[\underline{\xi},d(n)]|)}{d(n) \in Obj(|N|)}$$

$$\frac{}{|b[\underline{\xi},n,\underline{\zeta}]| \in Obj(|A[\underline{\xi},(s;d(n))]|)}$$

$$\frac{|a[\underline{\xi}]| \in Obj(|A[\underline{\xi},\underline{d(0)}]|) \qquad = Obj(|A[\underline{\xi},\underline{d((s;n))}]|)}{(rec(|a[\xi]|,|b[\underline{\xi},n,\underline{\zeta}]|);\underline{\xi},e(n)) \in Obj(|A[\underline{\xi},\underline{d(e(n))}]|)}$$

$$= |(rec(a[\underline{\xi}],b[\underline{\xi},x,y]);\underline{\xi},\underline{n})| \qquad = Obj(|A[\underline{\xi},\underline{n}]|)$$

N-conversion.

$|(rec(a[\underline{\xi}],b[\underline{\xi},x,y]);a_1,0)| = (rec(|a[\underline{\xi}]|,|b[\underline{\xi},n,\underline{\zeta}]|);|a|,e(|0|))$

$= (rec(|a[\underline{\xi}]|,|b[\underline{\xi},n,\underline{\zeta}]|);|a_1|,0)$ conv $|a[\underline{|a_1|}]| = |a[a_1]|$

$|(rec(a[\underline{\xi}],b[\underline{\xi},x,y]);a_1,(s;c))| = (rec(|a[\underline{\xi}]|,|b[\underline{\xi},n,\underline{\zeta}]|);|a_1|,e(|(s;c)|))$

$= (rec(|a[\underline{\xi}]|,|b[\underline{\xi},n,\underline{\zeta}]|);|a_1|,(s;|c|))$ conv

$|b[\underline{|a_1|},e(|c|),\underline{(rec(|a[\underline{\xi}]|,|b[\underline{\xi},n,\underline{\zeta}]|);|a_1|,|c|)}]|$

$= |b[a_1,c,(rec(a[\underline{\xi}],b[\underline{\xi},x,y]);a_1,c)]|$

Group D

i) $p(|\Sigma(A[\underline{\xi}],B[\underline{\xi},x])|) =_{df} 0$

$q(|\Sigma(A[\underline{\xi}],B[\underline{\xi},x])|)$ is the species of realizing objects given by:

$q(|\Sigma(A[\underline{\xi}],B[\underline{\xi},x])|)(\alpha) = \Sigma\zeta \in Obj(|A[\underline{\xi}]|).\Sigma\chi \in Obj(|B[\underline{\xi},\zeta]|)$

$\alpha = (p(\zeta),p(\chi))$

ii) $p(|(pr;\underline{\xi},a[\underline{\xi}],b[\underline{\xi}])|) =_{df} [p(|a[\underline{\xi}]|),p(|b[\underline{\xi}]|)]$

$q(|(pr;\underline{\xi},a[\underline{\xi}],b[\underline{\xi}])|)$ is the proof that it is in the species

$q(|\Sigma(A[\underline{\xi}],B[\underline{\xi},x])|)$, and is defined as follows:

$q(|(pr;\underline{\xi},a[\underline{\xi}],b[\underline{\xi}])|) =_{df} (|a[\underline{\xi}]|,(|b[\underline{\xi}]|,r([p(|a[\underline{\xi}]|),p(|b[\underline{\xi}]|)])))$

$\|(pr;\underline{\xi},a[\underline{\xi}],b[\underline{\xi}])\|(\alpha) =_{df} [\|a[\underline{\xi}]\|(\alpha),\|b[\underline{\xi}]\|(\alpha)]$, from 4)ii)a)

iii) $|(un(c[\underline{\xi},x,y]);\underline{\xi},\underline{\zeta})| =_{df} (un((un(|c[\underline{\xi},\underline{\chi},\underline{\tau}]|);\underline{\xi},\chi,\nu));\underline{\xi},q(\zeta))$

$\|(un(c[\underline{\xi},x,y]);\underline{\xi},\underline{\zeta})\|(\alpha,\beta) =_{df} \|c[\underline{\xi},x,y]\|(\alpha,p(\beta),q(\beta))$

$\underline{\Sigma\text{-reflection}}$. Similar to group B.

$\underline{\Sigma\text{-introduction}}$. Obvious.

$\underline{\Sigma\text{-elimination}}$. Assume that for $\chi \in Obj(|A[\underline{\xi}]|)$ & $\tau \in Obj(|B[\underline{\xi},\underline{\chi}]|)$

$|c[\underline{\xi},\underline{\chi},\tau]| \in Obj(|C[\underline{\xi},(pr;\underline{\xi},\underline{\chi},\underline{\tau})]|) = Obj(|C[\underline{\xi},\underline{|(pr;\underline{\xi},\underline{\chi},\underline{\tau})|}]|)$

$= Obj(|C[\underline{\xi},\underline{([p(\chi),p(\tau)],(\chi,(\tau,r([p(\chi),p(\tau)])))}]|)$

Then for $\zeta \in Obj(|\Sigma(A[\underline{\xi}],B[\underline{\xi},x])|)$:

if $w \in p(\zeta) = [p(\chi),p(\tau)]$ then

$|c[\underline{\xi},\underline{\chi},\underline{\tau}]| \in Obj(|C[\underline{\xi},\underline{(p(\zeta),(\chi,(\tau,r(p(\zeta)))))}]|)$

$= Obj(|C[\underline{\xi},\underline{(p(\zeta),(\chi,(\tau,w)))}]|)$ by 1.4.6.

and if $v \in \Sigma\tau \in Obj(|B[\underline{\xi},\underline{\chi}]|).p(\zeta) = [p(\chi),p(\tau)]$ then

$(un(|c[\underline{\xi},\underline{\chi},\underline{\tau}]|);\underline{\xi},\chi,v) \in Obj(|C[\underline{\xi},\underline{(p(\zeta),(\chi,v))}]|)$

and if $u \in \Sigma\chi \in Obj(|A[\underline{\xi}]|).\Sigma\tau \in Obj(|B[\underline{\xi},\underline{\chi}]|).p(\zeta) = [p(\chi),p(\tau)]$ then

$(un((un(|c[\underline{\xi},\underline{\chi},\underline{\tau}]|);\underline{\xi},\chi,v));\underline{\xi},u) \in Obj(|C[\underline{\xi}, \underline{(p(\zeta),u)}]|)$

Hence:

$(un((un(|c[\underline{\xi},\underline{\chi},\underline{\tau}]|);\underline{\xi},\chi,v));\underline{\xi},q(\zeta)) \in Obj(|C[\underline{\xi}, \underline{(p(\zeta),q(\zeta))}]|)$

$= Obj(|C[\underline{\xi},\underline{\zeta}]|)$ by 1.4.4.

Σ-conversion.

$|(un(c[\underline{\xi},x,y]);a_1,(pr;a,b))|$ conv

$(un(|c[\underline{\xi},\underline{\chi},\underline{\tau}]|);a_1,|a|,(|b|,r((p(|a|),p(|b|)))))$

conv $|c[a_1,\underline{|a|},\underline{|b|}]| = |c[a_1,a,b]|$

Group E

i) $|(=;\xi,a[\underline{\xi}],b[\underline{\xi}])| =_{df} (0,\lambda\alpha.\alpha=p(|a[\underline{\xi}]|) \ \& \ |a[\underline{\xi}]|=|b[\underline{\xi}]|.)$

ii) $|(r;\underline{\zeta})| =_{df} (p(\zeta),(r(p(\zeta)),r(\zeta)))$

$\|(r;\underline{\zeta})\| =_{df} p(\zeta)$

iii) $|(id(c[\underline{\xi},x]);\underline{\xi},\underline{\zeta})| =_{df} (id(|c[\underline{\xi},\underline{\chi}]|);\underline{\xi},q(q(\zeta)))$

$\|(id(c[\underline{\xi},x]);\underline{\xi},\underline{\zeta})\|(\alpha,\beta) =_{df} \|c[\underline{\xi},\underline{\chi}]\|(\alpha,\beta)$

=-reflection.

$Obj(|A[\underline{\xi}]|) \in V_k, D \in V_o$, $q(|a[\underline{\xi}] = b[\underline{\xi}]|) \in D \to V_k$; hence result.

=-introduction. Obvious.

=-elimination. Assume that for $\chi \in Obj(|A[\underline{\xi}]|)$,

$|c[\underline{\xi},\underline{\chi}]| \in Obj(|C[\underline{\xi},\underline{\chi},\underline{\chi},(r;\underline{\xi},\underline{\chi})]|)$

$= Obj(|C[\underline{\xi},\underline{\chi},\underline{\chi}, \underline{(p(\chi),(r(p(\chi)),r(\chi)))}]|)$

if $w \in |a[\underline{\xi}]| = |b[\underline{\xi}]|$ then:

$(id(|c[\underline{\xi},\underline{\chi}]|);\underline{\xi},w) \in Obj(|C[\underline{\xi},\underline{|a[\underline{\xi}]|},\underline{|b[\underline{\xi}]|},\underline{(p(|a[\underline{\xi}]|),(r(p(|a[\underline{\xi}]|)),r(|a[\underline{\xi}]|)]}$

So for $\zeta \in Obj(|(=;\underline{\xi},a[\underline{\xi}],b[\underline{\xi}])|)$, $\zeta=(p(|a[\underline{\xi}]|),(r(p(|a[\underline{\xi}]|)),r(|a[\underline{\xi}]|)))$

and hence: $(id(|c[\underline{\xi},\underline{\chi}]|);\underline{\xi},q(q(\zeta))) \in Obj(|C[\underline{\xi},a[\underline{\xi}],b[\underline{\xi}],\underline{\zeta}]|)$

<u>=-conversion</u>. $|(id(c[\underline{\xi},x]);a_1,(r;a))|$ conv $|c[a_1,\underline{|a|}]| = |c[a_1,a]|$

Chapter 3

RELATIVE CONSISTENCY

We might extend M-L in several ways. We could add a new typesymbol to the language, by associating some 0-ary function constant, F, with one of the V_i. And then changing the definition of $\in$ and conv, so that there is a clause: $F \in V_i$. Or given a typesymbol A, added in this way or not, we could add a closed term of that typesymbol, by associating an 0-ary function constant f with A and adding the clause: $f \in A$. In this case, an axiom has been added to M-L. Again, where A is not intended to be a proposition, we might add a whole family of members of A by an n-ary function constant of value typesymbol A, and both A-introduction and A-elimination clauses. Or in a similar way an inductively defined predicate. Again instead of adding the axiom $A \to B$, we might add the rule $\frac{A}{B}$. Each of these extensions relate to this chapter, but we shall be mainly concerned with the relative consistency of adding an axiom A or a rule $\frac{A}{B}$ to M-L.

In the first case relative consistency is established by proving, for some model M of M-L, that $Obj(|A|)$ is not empty. For then we could set $|a| = a'$ if $a' \in Obj(|A|)$; thus giving a model M' for M-L and A. In the second case we find a model of M-L and the rule $\frac{Obj(|A|)}{Obj(|B|)}$. At the same time, relative consistency is proved only if we can also show that $Obj(\bot)$ is empty i.e. that $Obj(|0=1|) \to \bot$. This is guaranteed in realizability models. For we have $\mathbb{0} \neq \$(\mathbb{0})$ so $\mathbb{0} \neq \mathbb{1}$ and hence $|0| \neq |1|$.

§1. Candidates for relative consistency (by realizability models)

When looking at relative consistency results for systems of intuitionistic analysis, it is natural to ask what principles of continuity hold. In some ways it is the motivation behind realizability models to validate such constructive principles. Also Church's Thesis is an intuitive expression of constructivity one would hope to verify. But before we can put these to

the test, formal counterparts in the theory must be chosen. These are typesymbols, whose intended interpretations are those principles. There may be more than one formal counterpart, and we begin by investigating the possible bifurcations.

In this chapter, it will be convenient to refer to finite types by their level in the hierarchy. Thus N will be abbreviated by 0, $N \to N$ by 1, $(N \to N) \to N$ by 2, $(N \to N) \to (N \to N)$ by $1 \to 1$, etc. Also we shall use the convention of assuming that the variables n,m,k,... are of type N; that f,g,h,... are of type $N \to N$ and F,G,H,... are of types $(N \to N) \to N$ or $(N \to N) \to (N \to N)$ depending on whether they are superscripted 2 or $1 \to 1$. And, further, we use the abbreviation '$\forall n < m.$' for '$\forall n.n < m$ ' and '$\exists n < m.$' for '$\exists n.n < m \;\&$' α '$f \in g(n)$' for '$\forall m < n.f(m)=g(m).$'

3.1.1. Continuity principles

Taking Kreisel-Howard's paper [1966] as a starting point, we have the schema of weak continuity at type 2:

$$\text{cont}_R(2) =_{df} \forall f.\exists n.R(f,n) \to \forall f.\exists n,m.\forall g.f \in \bar{g}(n) \to R(g,m).$$

and the strong schema

$$\text{s.cont}_R(2) =_{df} \forall f.\exists n.R(f,n) \to \exists g.\forall f.(\exists n.g(\bar{f}(n)) \neq 0$$
$$\& \forall c,x.g(c) \neq 0 \to g(c\,x) \neq 0.$$
$$\& \forall m.g(\bar{f}(m)) \neq 0 \to R(f,g(\bar{f}(m))-1).)$$

Which is Kleene-Vesley's 'Brower's principle for numbers', and is discussed in [1965] page 79. However, it should be noted that, given the intended interpretation of the symbols of H-L, it is stronger than Brouwer's principle. Consequently, any relationships that we establish between this schema and other axioms of H-L , cannot be said to hold of Brouwer's principle as such. They will be true only within H-L.

We make use of higher type variables to give the axioms:

$$\text{cont}(2) =_{df} \forall F^2.\forall f.\exists n.\forall g.f \in \bar{g}(n) \to F(f)=F(g).$$

$$\text{s.cont}(2) =_{df} \forall F^2.\exists g.\forall f.\exists n.\forall n' \geq n.g(\bar{f}(n')) = F(f) + 1$$
$$\& \forall m<n.g(\bar{f}(m)) = 0$$

And whilst it is clear that the strong schema and axiom are really stronger than their counterparts; for example:

3.1.1.1. $\vdash s.cont_R(2) \to cont_R(2)$

Proved by taking the m to be $g(\bar{f}(n)) \dot{-} 1$.

The schema and the corresponding axiom are equivalent:

3.1.1.2. $\vdash cont(2) \leftrightarrow cont_R(2)$

$\to$. Applying the axiom of choice to $\forall f.\exists n.R(f,n)$ gives us $\exists F^2.\forall f.R(f,F(f))$. Then for this F, from the cont(2) we have

$$\forall f.\exists n.\forall g.f \in \bar{g}(n) \to F(f)=F(g).$$

and $\forall g.R(g,F(g))$. Combining, and taking $m = F(f)$ gives result.

$\leftarrow$. Fix an $F \in 2$, and put $R(f,n) =_{df} F(f)=n$. Then

$$\forall f.\exists n.F(f)=n$$

and so from $cont_R(2)$:

$$\forall f.\exists n,m.\forall g.f \in \bar{g}(n) \to F(g)=m,$$

and $m=F(f)$, as $f \in \bar{f}(n)$. Hence results.

3.1.1.3. $\vdash s.cont(2) \leftrightarrow s.cont_R(2)$

$\to$. Use the axiom of choice. $\leftarrow$. Put $R(f,n) =_{df} F(f)=n$.

Now, the thought behind continuity at type two is that the value of the function can be calculated from a finite amount of information about the argument. We can simply extend this to objects of type $1\to1$; they are continuous just in case a finite amount of information about the value can be derived from finite information about the argument. Hence we have a weak axiom:

$$cont\ (1\to1) =_{df} \forall F^{1\to1}.\forall f,n.\exists m.\forall g.f \in \bar{g}(m) \to F(f) \in \overline{F(g)}(n).$$

and strong schema and axiom:

$$s.cont_R(1\to1) =_{df} \forall f.\exists g.R(f,g) \to \exists h.\forall f.\ h|f \text{ exists} \ \& \exists k.k =_{ext} h|f \ \& \ R(f,k).$$

which is Kleene's 'Brower's Principle', and

$$s.cont(1\to1) =_{df} \forall F^{1\to1}.\exists h.\forall f.h|f \text{ exists } \& \ F(f) =_{ext} h|f$$

where $h|f$ exists $=_{df} \forall t\exists n.\, h(t*\bar{f}(n)) \neq 0$ and '$|$' is defined in section 2.4.2 above.

Again there is no bifurcation between the schema and the axiom:

3.1.1.4. $\vdash s.cont(1\to1) \leftrightarrow s.cont_R(1\to1)$

$\to$. Use the axiom of choice

$\leftarrow$. Put $R(f,g) = F(f)=g$

But more substantially, the type 1→1 principles turn out to be equivalent to the type 2 ones:

3.1.1.5. $\vdash cont(2) \leftrightarrow cont(1\to1)$

$\leftarrow$. Given $F \in 2$ define $F' \in 1\to1$ by:

$$F'(\alpha)(n) =_{df} F(\alpha)$$

and fixing n=0 (say) the result is immediate.

$\to$. Given $F \in 1\to1$ and $n \in N$ define $F_t \in 2$ by:

$$F_t(\alpha) =_{df} \overline{F(\alpha)}(t) \text{ for each } t \leq n$$

then fixing f let m be $\max_{t\leq n}(N(F_t,f))$, where N is the modulus of continuity functional, obtained from cont(2) by two applications of the axiom of choice.

3.1.1.6. $\vdash s.cont(2) \leftrightarrow s.cont(1\to1)$

$\leftarrow$. For $F \in 2$ define $F' \in 1\to1$ by:

$$F'(\alpha) =_{df} \lambda x.F(\alpha)$$

So , $\exists h.\ \forall f.\ \exists n.$

$$h\langle 0*\bar{f}(n)\rangle \neq 0$$

and $h\langle 0*\bar{f}(\mu y.h\langle 0*\bar{f}\langle y\rangle\rangle) = F'(f)(0) + 1$

$\neq 0$

Hence define

$$g =_{df} \lambda x.h\langle 0*x\rangle \dot{-} 1$$

Then , $\forall f.\ \exists n\ (= \mu y.h\langle 0*\bar{f}(y)) \neq 0)$

and $g\langle \bar{f}(n)) = h\langle 0*\bar{f}\langle n)) \dot{-} 1 = F'(f)\langle 0) = F(f)$

and $\forall n' < n.\ g\langle f(n')) = 0$

$\rightarrow$. For $F \in 1\rightarrow 1$ and $t \in N$ define $F_t \in 2$ by:

$$F_t(\alpha) =_{df} F(\alpha)(t)$$

Then $\exists g_t.\ \forall f.\ \exists n.$ (*)

$$g_t(\bar{f}\langle n)) = F_t(f)(t) + 1$$

and $\forall n' < n.\ g\langle \bar{f}(n')) = 0$

So define h by:

$$h(c) =_{df} g_{c_c}(\langle c_1\ c_2 \ldots c_{lh(c)\dot{-}1}\rangle)$$

Then $\forall f,t.\ \exists m$ (= the n of (*)) such that

$$g_t(\bar{f}(m)) = F_t(f) + 1$$

$$h\langle t*\bar{f}\langle m)) = F_t(f) + 1$$

and $m = \mu y.g\langle \bar{f}(y)) \neq 0$

hence

$$h\langle t*\bar{f}\langle n)) \neq 0 \quad \text{and} \quad h|f\langle t) = F_t(f) = F(f)(t)$$

i.e. $h|f$ exists and $h|f =_{ext} F(f)$.

We shall now prove a relative consistency result:

3.1.1.7. Theorem. S.cont(2) is consistent relative to M-L

This axiom is validated by the set realizability model. We shall defer the proof to §6; where the method will be to give a generalized

definition of strong continuity over the whole typestructure, which is s.cont(2) at type 2. The set realizability model will be used there to validate an axiom of strong continuity at all types.

3.1.1.8. Corollary. Cont(2) is consistent.

There is a proof of the relative consistency of s.cont(2) with HA, which is to be found e.g. in Kleene-Vesley [1965] p. 109. And it uses function realizability. However we cannot carry the proof over to M-L; for there they have a number theoretic function being realized by itself, and this is not so for us. In fact, if number theoretic functions are so realized, then the proof of the consistency of s.cont(2) is trivial.

We might attempt to validate cont(2) directly in the set realizability model. It would proceed as follows: given a $\chi \in Obj(|2|)$ and $\xi \in Obj(|1|)$ consider $p(\chi)|p(\xi)$. This will be a singleton set, $\{t\}$, from which we can find an n such that $e_n \subseteq p(\xi)$ and $(n,t) \in p(\chi)$. Then clearly, for any $\zeta \in Obj(|1|)$ if $e_n \subseteq p(\zeta)$ then $p(\chi)|p(\zeta) \simeq p(\chi)|p(\xi)$. And each m with $m \in e_n$ and so $\in p(\xi)$ determines a k_m such that $p(\xi)|p(|k_m|) \simeq \{m\}$. And taking the maximum of these k's, K, we have that if for every $r \leq K$ $p(\xi)|p(|r|) \simeq p(\zeta)|p(|r|)$ then $e_n \subseteq p(\zeta)$. However, in order to verify the axiom we must show that $p(|K|)$ is continuous in $p(\chi)$ and $p(\xi)$, so that Λ-abstraction will give a member of $Obj(|cont(2)|)$. However the $p(|K|)$ is not continuous in these. What is validated by this argument, though, is the rule:

$$\frac{\text{F is a closed term of type 2, f of type 1}}{\exists n.\forall g.f \in \bar{g}(n) \rightarrow F(f) = F(g)\ .}$$

3.1.2. Church's Thesis

This is another candidate for relative consistency. A formal counterpart of Church's Thesis at type 1 is:

$$chth(1) =_{df} \forall f.\exists e.\forall n.\exists m.T(e,n,m) \ \&\ \forall m'<m.\sim T(e,n,m') \ \&\ f(n) = U(m).$$

where T and U are the M-L formalizations of the usual, predicate and function, respectively. And we write $f \underset{ext}{=} \{e\}$.

From now on, where the corresponding schema is obviously equivalent, we do not state it. But a simple observation shows that chth(1) and cont(2) are mutually inconsistent.

3.1.2.1. $\vdash\sim$(chth(1) & cont(2)).

Using the axiom of choice on chth(1) and applying cont(2) to the derived functional, we get that the gödel-number of a function f is dependent on only its first n_f (say) places. So define $f' \in N \to N$ by:

$$f'(p) = \begin{cases} f(p) & \text{if } p \leq n_f \\ f(p)+1 & \text{o.w.} \end{cases}$$

then f and f' have the same gödel-number, e (say), and so $f(n_f+1) = \{e\}(n_f+1) = f'(n_f+1) = f(n_f+1) = 1$; contradiction.

Furthermore, higher type axioms of recursiveness have cont(2) as a consequence, and so e.g. $\sim$(chth(1) & chth(2)), where

$$\text{chth}(2) =_{df} \forall F^2.\exists e.\forall f.\exists n,m.T(e,\bar{f}(n),m) \;\&\; F(f) = U(\mu_m T(e,\bar{f}(n),m)).$$

$$\text{chth}(1\to1) =_{df} \forall F^{1\to1}.\exists e.\forall f,p.\exists n,m.T(e,\bar{f}(n),m) \;\&\; F(f)(p) = U(\mu_m T(e,\bar{f}(n),m)).$$

and as before we could show that $\vdash$chth(2) $\leftrightarrow$ chth(1→1).

To show this we prove that chth(2) has the stronger consequence of s.cont(2):

3.1.2.2. $\vdash$chth(1→1) → s.cont(1→1)

Just formalize Kleene's lemma (K.V.p.90), which states that $\forall F^{1\to1}$. F is recursive → $\exists h.\forall f.h|f$ exists & $F(f) \underset{ext}{=} h|f$.

We shall now prove another relative consistency result:

3.1.2.3. Theorem. Chth(1) is consistent with M-L.

We use the number realizability model. Given $\zeta \in Obj(|N \to N|)$ then $p(\zeta)$ is such that $\Pi\eta \in Obj(|N|)$.

$$\{p(\zeta)\}(p(n)) = p(|\underline{\zeta}(\underline{n})|)$$

and $\Sigma m \in N.T(p(\zeta),p(\eta),m) \;\&\; \forall m' < m \sim T(p(\zeta),p(\eta),m') \;\&\; p(|\underline{\zeta}(\underline{n})|) = U(m)$

Hence we have a $\mu \in Obj(|N|)$, μ =(m,proof of $\mathbb{N}(m)$) and

$$T(p(\zeta),p(\eta),p(\mu)) \quad \& \quad p(|\underline{\zeta}(\underline{n})|) = U(p(\mu))$$

and as T,U are primitive recursive:

$$Obj(|T(\underline{\zeta},\underline{n},\underline{\mu})|) \;\&\; p(|\underline{\zeta}(\underline{n})|) = p(|U(\underline{\mu})|)$$

$$Obj(|T(\underline{\zeta},\underline{n},\underline{\mu})|) \;\&\; Obj(|\underline{\zeta}(\underline{n}) = U(\underline{\mu})|)$$

So we now have: $\Pi\zeta \in Obj(|N \to N|)$. $\Sigma\xi \in Obj(|N|)$. ξ =(p(ζ), proof of $\mathbb{N}(p(\xi))$) $\Pi\eta \in Obj(|N|)$. $\Sigma\mu \in Obj(|N|)$. $Obj(|T(\underline{\zeta},\underline{n},\underline{\mu})|) \;\&\; Obj(|\underline{\zeta}(\underline{n})=U(\underline{\mu})|)$. and to get an object of type $Obj(|chth(1)|)$ we need only show that $p(\xi)$ is recursive in $p(\zeta)$ (use the identity function) and that $p(\mu)$ is recursive in $p(\zeta),p(\eta)$ which is true as $p(\mu) = \mu_m T(p(\zeta),p(\eta),m))$. And the result follows by S^n_m-theorem.

3.1.2.4. Corollary. Cont(2) is independent; and hence is s.cont(2), chth(2) etc. as they are inconsistent with chth(1).

§2. Other models over an r.s.

What I have called the model over an r.s. is not the only possible mdoel, and it will be occasionally useful to alter the superstructure. We shall construct other models over an r.s.:

3.2.1. By making $p(|A|)$ unique, up to $\simeq$; as follows

$$p(|N|) =_{df} [\mathbb{0},\mathbb{0}]$$

$$p(|V_n|) =_{df} [\mathbb{1},p(|\mathbf{n}|)]$$

$$p(|\Pi(A,B[x])|) =_{df} [2,[p(|A|),\lambda\alpha.\|B[x]\|(\alpha)]]$$

$$p(|\Sigma(A,B[x])|) =_{df} [3,[p(|A|),\lambda\alpha.\|B[x]\|(\alpha)]]$$

$$p(|(=;a,b)|) =_{df} [4,[p(|C|)[p(|a|),p(|b|)]]]$$

Then we can prove, for example:

Theorem. The rule $\dfrac{\text{for closed terms } A,B \text{ of } V_n}{A = B + A \neq B}$ is consistent.

To prove this, use any r.s. on whose domain $\simeq$ is decidable, and make the above alterations.

3.2.2. By altering $q(|V_n|)$; whilst retaining the requirement that $Obj(|V_n|) = Typ_n$.

The method is best demonstrated by using it to prove:

Theorem. For each n, $\sim$Count(V_n) is consistent relative to M-L.

We can use any r.s. on whose domain = is decidable. And use the decidability to define $q(|V_n|)$ by cases.

We show first that: $\Pi f \in V_n \to N.\Sigma m \in N.\Pi X \in V_n.f(X) = m$ is consistent by making $|V_n| =_{df} (0,\lambda\alpha.f_n(\alpha))$ where

$$f_n(\alpha) =_{df} \begin{cases} D \to V_{n-1} & \text{if } \alpha = 0 \\ \\ \perp \times V_{n-1} & \text{o.w.} \end{cases}$$

and $Obj(|V_n|) = D\times(D \to V_{n-1}) = Typ_n$.

Then $\Pi\chi \in Obj(|V_n|).p(\chi) = \mathbb{0}$; as $\mathbb{0} = p(\chi) + \mathbb{0} \neq p(\chi)$ and if $\mathbb{0} \neq p(\chi)$ then $q(\chi) \in f_n(p(\chi)) = \perp \times V_{n-1}$

and $pq(\chi) \in \perp$

So for $\zeta \in Obj(|V_n \to N|)$ let $n =_{df} |\underline{\zeta}(V_{n-1})| \in Obj(|N|)$

and then for all $\chi \in Obj(|V_n|)$

$$p(|\underline{\zeta}(\chi)|) = p(\zeta)|p(\chi) = p(\zeta)|\Phi$$

$$= p(\zeta)|p(|V_{n-1}|) = p(|\underline{\zeta}(V_{n-1})|) = p(\eta)$$

hence $Obj(|\underline{\zeta}(\chi) = \underline{\eta}|)$ and result follows.

Now, $\Pi f \in V_n \rightarrow N.\Pi X,Y \in V_n.\ f(X)=f(Y)$ and so:

$$\Sigma f \in V_n \rightarrow N.\Sigma g \in N \rightarrow V_n.\Sigma X \in V_n.\ g_o f(X)=X \rightarrow \Pi X,Y \in V_n.\ X=Y$$

$$\rightarrow V_{n-1} = \sim V_{n-1}$$

$$\rightarrow (V_{n-1} \rightarrow \sim V_{n-1}) \ \& \ (\sim V_{n-1} \rightarrow \sim\sim V_{n-1})$$

$$\rightarrow \sim V_{n-1} \ \& \ \sim\sim V_{n-1}$$

$$\rightarrow \perp$$

Hence $s.count(V_n) \rightarrow \perp$.

3.2.3. By submodels.

We could give a general account of what it is to be a submodel of M-L, but it will be more profitable to concentrate on submodels of realizability models of M-L. The reason for this is; suppose we have a model and want to reduce it, that is reduce the membership of the type that interprets a typesymbol qua typesymbol in the model. If A is the typesymbol and we have $Obj(|A|)$ in the original model, then the reduction would be made by defining a species, Φ_A, of members of $Obj\ (|A|)$. The new interpretation of A qua typesymbol would be the typesymbol $\Sigma\xi \in Obj(|A|)$. $\Phi_A(\xi)$ and Φ would be defined over the typestructure of M-L. The new interpretation of A qua term would be the pair $(|a|$, proof of $\Phi_A(|a|))$.

Such a uniform procedure would produce a kind of submodel, but it would be useless for consistency result applications. To see this we need only note that given a typesymbol, P, which is the formal counterpart of some putative consistent axiom; if c is a constant of the type that interprets P qua typesymbol in the submodel then part of c is of type $Obj(|P|)$ and so P

is validated by the original model. We may however be able to show that the interpretation of P qua typesymbol is empty in the submodel, whereas Obj(|P|) may not be. That is, we may be able to use the procedure for independence results.

By concentrating on realizability models, we can define the species Φ_A at the same time as we define the species which is part of the interpretation of A qua term. In this way the interpretation of a typesymbol qua typesymbol in the submodel will depend on the interpretation of its sub-terms which are typesymbols. This is not a uniform procedure and so will not lead to the difficulties described in the previous paragraph. Such submodels I distinguish by the term 'realizability submodels' and are referred to the r.s. over which they are constructed. A realizability submodel over an r.s. is given by:

i) $\mathrm{Typ}_k^* =_{df} \Sigma X \in \mathrm{Typ}_k.\mathrm{Obj}(X) \to V_k$

ii) for each $\phi \in \mathrm{Typ}_k^*$

$\mathrm{Obj}_k^*(\phi) =_{df} \Sigma\xi \in \mathrm{Obj}_k(p(\phi)).q(\phi)(\xi)$

iii) $|.|^*$ is defined so that for $a \in A$

$|a|^* \in \Sigma\xi \in \mathrm{Obj}(p(|A|^*)).\Phi_A(\xi)$

where $\Phi_A =_{df} q(|A|^*)$

and $\|.\|^*$ is defined so that for $b[x_1,\ldots,x_n] \in B[x_1,\ldots,x_n]$

$\|b[x_1,\ldots,x_n]\|(p(p(\xi_1)),\ldots,p(p(\xi_n))) \simeq p(p(|b[\underline{\xi_1},\ldots,\underline{\xi_n}]|^*))$

To demonstrate the method we give the definitions of $|.|^*$, $\|.\|^*$ for the group B rules; the other cases are similar.

i) $p(p(|\Pi(A[\underline{\xi}^*],B[\underline{\xi}^*,x])|^*) =_{df} \emptyset$

$q(p(|\Pi(A[\underline{\xi}^*],B[\underline{\xi}^*,x])|^*)(\alpha) =_{df} \Pi\zeta^* \in \mathrm{Obj}^*(|A[\underline{\xi}^*]|^*).\Sigma\chi^* \in \mathrm{Obj}^*(|B[\underline{\xi}^*,\underline{\zeta}^*]|^*).$

$\alpha|p(p(\zeta^*)) \simeq p(p(\chi^*)) \;\&\; C(\alpha,p(p(\zeta^*)))$

$q(|\Pi(A[\xi^*],B[\xi^*,x])|^*) =_{df} \Phi_{\Pi(A,B)}$

ii) $p(p(|\lambda b[\xi^*,x]|^*) =_{df} \lambda\alpha.\|b[\xi^*,x]\|(p(p(\xi^*)),\alpha)$

$q(p(|\lambda b[\xi^*,x]|^*)) =_{df} \lambda\zeta^*.(|b[\xi^*,\zeta^*]|^*,(p_1^*,p_2^*))$

where p_1^*,p_2^* are the obvious extensions of the objects 2.5 above.

$q(|\lambda b[\xi^*,x]|^*)$ = the proof that $\Phi_{\Pi(A,B)}(p(|\lambda b[\xi^*,x]|^*))$.

$\|b[\xi,x]\|^*(\beta) =_{df} \lambda\alpha.\|b[\xi,x]\|^*(\beta,\alpha)$

iii) $|(ap;\xi^*,f[\xi^*],a[\xi^*])|^* =_{df} p(q(|f[\xi^*]|^*)(|a[\xi^*]|^*)$

$\|(ap;\xi^*,f[\xi^*],a[\xi^*])|^*(\alpha) = \|f[\xi^*]\|(\alpha)\,\big|\,\|a[\xi^*]\|(\alpha)$

and the proofs are exemplified by Π-introduction, where if for $\zeta^* \in Obj(|A[\xi^*]|^*)$, $|b[\xi^*,\zeta^*]|^* \in Obj^*(|B[\xi^*,\zeta^*]|^*)$ then as in 2.5 above

$p(|\lambda b[\xi^*,x]|^*) \in Obj^*(p(|\Pi(A[\xi^*],B[\xi^*,x])|^*))$

and the proof is completed by showing that

$\Phi_{\Pi(A,B)}(p(|\lambda b[\xi^*,x]|^*))$

In future we shall give the submodel by defining the species and proving that for $a \in A$, $\Phi_A(p(|a|^*))$.

§3. Two candidates for relative consistency (By submodels)

3.3.1. Extensionality

The functions of types $A \to B$, $\Pi x \in A,B[x]$ are not strictly maps; they take (in general) only identical objects to identical objects not extensional to extensional. For example, take A and B to be the domain of the set realizability model, and consider

$F \in (N \to N + T) \to (N \to N + T)$, defined by:

$$F(\alpha) =_{df} \lambda x.\alpha(0)$$

and $\omega \in N \to N + T$ by:

$$\omega(x) =_{df} \begin{cases} i(1) & \text{if } x=0 \\ i(x-1) & \text{o.w.} \end{cases}$$

then, $F(i) = \{0\}$ and $F(\omega) = \{1\}$.

So we have $\Pi x \in N.\ x \varepsilon i \leftrightarrow x \varepsilon \omega \ \&\ F(i) \neq F(\omega)$. Where i is the injection of N into N+T.

In some sense F is not a map.

Thinking of $N \to N + T$ simply as a type of functions, however, it is not clear whether I can find a pair of functions $f,g \in N \to N + T$ with $\Pi x \in N.f(x) = g(x)$ but not $\Pi x \in N.F(f)(x) = F(g)(x)$. There is therefore no hope of saying what we mean by an extensional object of an arbitrary type unless we restrict attention to a particular interpretation. For this reason we start with the higher types where the intended interpretation is unambiguous: we define, what it is for two objects of a higher type to be extensional:

If $n,m \in N$, then

$$n =_{ext} m =_{df} n=m$$

If $f,g \in A \to B$, then

$$f =_{ext} g =_{df} \Pi x \in A.f(x) =_{ext} g(x)$$

And what we ask is that all objects of a higher type are maps. We express this by a series of axioms:

$$ext(N) =_{df} T$$

$$ext(A \to B) =_{df} \Pi F \in A \to B.\Pi f,g \in A.f =_{ext} g \to F(f) =_{ext} F(g).$$

The first thing to note is that some of these axioms stated for arithmetic types reduce to those for finite types.

3.3.1.1. $\vdash ext(A \to (B \to C)) \leftrightarrow ext(A \to C)$

$\to$. Specialization; given $F \in A \to C$ define $F' \in A \to (B \to C)$ by

$$F'(x) =_{df} \lambda y.F(x)$$

$\leftarrow$. Given $F \in A \to (B \to C)$; $f,g \in A$; $f =_{ext} g$ and $k \in B$ define $F_k \in A \to C$ by:

$$F_k(h) =_{df} F(h)(k)$$

then we have

$$F_k(f) =_{ext} F_k(g)$$

$$F(f)(k) =_{ext} F(g)(k) \quad \text{so } F(f) =_{ext} F(g)$$

and by repetition we get $\vdash ext(A \to C) \leftrightarrow ext(A \to 0)$, so in particular $\vdash ext(n \to n) \leftrightarrow ext(n+1)$.

Another aspect of extensionality is whether extensionality and identity are the same thing at a particular type. There are axioms stating that the two relations collapse for each higher type:

$$col(A \to B) =_{df} \Pi f,g \in A \to B.f =_{ext} g \to f=g.$$

and that they are separated:

$$sep(A \to B) =_{df} \Sigma f,g, \in A \to B.f =_{ext} g \ \& \ f \neq g.$$

Then clearly:

3.3.1.2. $\vdash col(A) \to ext(A \to B)$

In fact $\vdash col(A) \to ext^*(A \to B)$ where:

$$ext^*(A \to B) =_{df} \Pi F \in A \to B.\Pi f,g \in A.f =_{ext} g \to F(f)=F(g).$$

3.3.1.3. $\vdash col(A \to B) \to col(B)$

Given $f,g \in B$ such that $f =_{ext} g$ then:

$$\lambda x.f, \lambda y.g \in A \to B \quad \text{and} \quad \lambda x.f =_{ext} \lambda y.g$$

so by col(A → B):

$$\lambda x.f = \lambda y.g$$

$$f = g$$

The real significance of col(A → B) is that it collapses bifurcations of the possible formal counterparts of some principles. Already, in formalizing the continuity and recursiveness principles we chose the weaker of possible alternatives. Here are some stronger versions:

$$s.cont^{*}_{R}(1\to1) =_{df} \forall f.\exists g.R(f,g) \to \exists h.\forall f.h|f \text{ exists } \& \ R(f,h|f).$$

$$s.cont^{*}(1\to1) =_{df} \forall F^{1\to1}.\exists h.\forall f.h|f \text{ exists } \& \ F(f) = h|f.$$

$$cthh^{*}(1) =_{df} \forall f.\exists e.\Sigma c \in \forall x.\exists y.T(e,x,y).\forall y' \leq y.\sim T(e,x,y')$$
$$\& \ f = \lambda x.U(p(c(x)))$$

and there are many more at higher types.

Using the following definitions:

$$dec(A) =_{df} \Pi x,y \in A.\ x =_{ext} y + \sim (x =_{ext} y)$$

$$s.dec(A) = \Pi x,y \in A.\ x=y + x\neq y$$

we can state many interrelated results.

3.3.1.4. ⊢chth(1) & ext(2) → dec(1)

Apply a.c.(1,0) to chth(1) to give the functional F which takes a function to its gödel-number. Then from ext(2) we have

$$\forall f,g.\ f =_{ext} g \leftrightarrow F(f)=F(g)$$

and as dec(N), we immediately get dec(1).

Similarly:

$$chth^{*}(1) \ \& \ ext(2) \to s.dec(1)$$

$$chth(2) \ \& \ ext(3) \to dec(2) \qquad \text{etc.}$$

3.3.1.5. $\vdash \text{cont}(2) \to \text{ext}(2)$

3.3.1.6. $\vdash \text{s.cont}^*(1\to1)\ \&\ \forall h.\forall f,g.\ f =_{ext} g \to h|f=h|g. \to \text{col}(1)$

The antecedents imply:

$$\forall F^{1\to1}.\forall f,g.\ f =_{ext} g \to F(f)=F(g)$$

and the result follows by putting $F =_{df} \lambda h.h$

3.3.1.7. $\vdash \sim (\text{cont}(2)\ \&\ \text{dec}(1))$

To see this define a type two object, F, using the decidability:

$$F(f) =_{df} \begin{cases} 0 & \text{if } f =_{ext} \lambda x.x \\ 1 & \text{o.w.} \end{cases}$$

then a $g \in N \to N$ defined by:

$$g(x) =_{df} \begin{cases} x & x \leq N(F,\lambda x.x) \\ x+1 & \text{o.w.} \end{cases}$$

is such that:

$$g \in \overline{\lambda x.x}(N(F,\lambda x.x))$$

but: $F(\lambda x.x) \neq F(g)$

3.3.1.8. $\vdash \sim (\text{ext}(2)\ \&\ \text{sep}(1)\ \&\ \text{s.dec}(1))$

Take $F \in 2$ defined by:

$$F(f') =_{df} \begin{cases} 0 & f' = \text{the f of sep}(1) \\ 1 & \text{o.w.} \end{cases}$$

then for the f,g of sep(1); F(f)=0 and F(g)=1 but $f =_{ext} g$

And the following independence and consistency results are known to follow:

3.3.1.9. ext(2) is consistent.

This follows from 3.3.1.5, and the consistency of cont(2).

3.3.1.10. the pair chth(1) and ext(2) are mutually independent.

Take 3.3.1.4., 3.3.1.7. and the consistency of cont(2), together.

3.3.1.11. sep(A) is consistent and col(A) is independent, for each A.

Use the term model where Obj(|f=g|) iff p(|f|) conv p(|g|), but where there are obviously extentionally equal non-convertable closed terms of all types e.g. for $N \to N$, $\lambda x.x-x$ and $\lambda x.0$.

3.3.1.12. ext(2) is independent.

Using the term model again; conv is decidable, so it is a model of both s.dec(1) and of sep(1) hence by 3.3.1.8. not of ext(2).

3.3.1.13. ext(3) is independent.

A counterexample can be made in the function realizability model. See Troelstra, §2.7.7, [1973].

We now make use of the realizability submodels to obtain further consistency results.

3.3.1.14. The extensional submodel of an r.s.

As mentioned at the end of 3.2.3., a realizability submodel is determined by the definition of Φ_A, simultaneous with that of $|.|^*$, so that $\Phi_A \in Obj(p(|A|^*)) \to V_n$ for $A \in V_n$. In the case we are about to describe we can determine the submodel by less. For each $A \in V_n$ define $\Psi_A \in D \to V_n$ and then produce the submodel as follows:

i) $Typ'_k =_{df} D \times (D \to V_k) \times (D \to V_k)$

ii) For each $\phi \in Typ'_k$:

$$Obj'_k(\phi) =_{df} \Sigma x \in D.\ p(q(\phi)'x) \ \&\ q(q'\phi)(x)$$

iii) the definition of $|.|'$ is such that $q(q(|A|')) =_{df} \Psi_A$. The result is just a special case of the models of 3.2.3. and so is described as a realizability submodel also.

In the particular case of the extensional submodel, we first define for each $A \in V_n$, $E_A \in D \times D \to V_n$ and then take Ψ_A to be the diagonal of E_A:

$$\Psi_A =_{df} \lambda x.E_A(x,x)$$

The motivation behind the definition of E being that:

$$E_n(p(\xi'),p(\zeta')) \Rightarrow Obj'(|\underline{\xi}' \underline{=}_{ext} \underline{\zeta}'|')$$

and $\Pi\chi' \in Obj'(|n|').\Psi_n(p(\chi')) \Rightarrow Obj'(|ext(n)|')$

The definition of E is simultaneous with that of $|.|'$, but we extract the clauses:

$$E_{V_n}(x,y) =_{df} D \to V_n$$

$$E_{\Pi x \in A.B[x]}(x,y) =_{df} \Pi\xi',\zeta' \in Obj'(|A|'). E_A(p(\xi'),p(\zeta')) \to E_{B[\underline{\xi}']}(x|p(\xi'),y|p(\zeta')).$$

$$E_N(x,y) =_{df} x=y$$

$$E_{\Sigma x \in A.B[x]}(x,y) =_{df} T$$

$$E_{a=b}(x,y) =_{df} T$$

In order to make the definition for $E_{C[x]}$ where the derivation of $C[x] \in V_n$ is obtained by an elimination rule , we need an induction hypothesis; as follows:
if B is a function constant of index/value typesymbols V_k / V_n then define a function

$$F_{E_B} \in (D \times D \to V_k) \to (D \times D \to V_n)$$

such that

$$F_{E_B}(E_A) = E_{B[A]}$$

Then we can define:

$$\mathrm{E}_{\mathrm{un}_B[x]} \quad =_{df} \quad \mathrm{un}_{\mathrm{E}_{B[p(x),q(x)]}}[x]$$

$$\mathrm{E}_{\mathrm{id}_B[x]} \quad =_{df} \quad \mathrm{id}_{\mathrm{E}_{B[a]}}[x]$$

$$\mathrm{E}_x \quad =_{df} \quad \lambda x.y.T$$

$$\mathrm{E}_{\mathrm{rec}_{A,B}[x]} \quad =_{df} \quad \mathrm{rec}_{\mathrm{E}_A,\mathrm{F}_{\mathrm{E}_B}}[x]$$

In fact, to complete the definition of $|.|'$ and prove that the interpretation of E and conv is standard, the induction hypothesis for open terms must be strengthened. If for

$$x_1 \mathrm{E} A_1, x_2 \mathrm{E} A_2[x_1], \ldots, x_n \mathrm{E} A_n[x_1,\ldots,x_{n-1}],\ b[x_1,\ldots,x_n] \mathrm{E} B[x_1,\ldots,x_n]$$

then in addition we require that:

for $\xi_1' \mathrm{E} \mathrm{Obj}'(|A_1|'),\ldots,\xi_n' \mathrm{E} \mathrm{Obj}'(|A_n[\xi_1',\ldots,\underline{\xi_{n-1}'}]|')$ and

for $\zeta_1' \mathrm{E} \mathrm{Obj}'(|A_1|'),\ldots,\zeta_{n-1}' \mathrm{E} \mathrm{Obj}'(|A_n[\zeta_1',\ldots,\underline{\zeta_{n-1}'}]|')$

$$E_{A_1}(p(\xi_1'),p(\zeta_1')) \;\&\; \ldots\ldots \;\&\; E_{A_n[\underline{\xi_1'}\,\underline{\xi_{n-1}'}]}(p(\xi_n'),p(\zeta_n')) \rightarrow$$

$$E_{B[\underline{\xi_1'}\,\underline{\xi_n'}]}(p(|b[\underline{\xi_1'},\ldots,\underline{\xi_n'}]|'),p(|b[\underline{\zeta_1},\ldots,\zeta_n']|'))$$

It will suffice to give the clauses of the definition of $|.|'$ just for group B:

i) $p(|\Pi(A[\underline{\xi}'],B[\underline{\xi}',x])|') =_{df} \Phi$

$p(q(|\Pi(A[\underline{\xi}'],B[\underline{\xi}',x]|')(\alpha) =_{df} \Pi\zeta' \in Obj'(|A[\underline{\xi}']|'). \Sigma\chi' \in Obj'(|B[\underline{\xi}',\underline{\zeta}']|').$

$\alpha|p(\zeta') \simeq p(\chi') \;\&\; C(\alpha,p(\zeta'))$.

$q(q(|\Pi(A[\underline{\xi}'],B[\underline{\xi}',x])|')) =_{df} \Psi_{\Pi(A[\underline{\xi}'],B[\underline{\xi}',x])}$

ii) $p(|\lambda b[\underline{\xi}',x]|') =_{df} \Lambda\alpha.\|b[\underline{\xi}',x]\|\ (p(\xi'),\alpha)$

$p(q(|\lambda b[\underline{\xi}',x]|') =_{df} \lambda\zeta'.(|b[\underline{\xi}',\underline{\zeta}']|',(p_1',p_2'))$

where p_1' and p_2' are derived as before.

$q(q(|\lambda b[\underline{\xi}',x]|') =_{df}$ the proof that $\Psi_{\Pi(A[\underline{\xi}'],B[\underline{\xi}',x])}(|\lambda b[\underline{\xi}',x]|')$ derived from the induction hypothesis.

iii) $|(ap;\ \underline{\xi}'\ ,f[\underline{\xi}'],a[\underline{\xi}'])|' =_{df} p(p(q(|f[\underline{\xi}']|'))(|a[\underline{\xi}']|'))$

3.3.1.15. Properties of the extensional submodel

Firstly, note that E is transitive. Prove this by induction on the definition of E, checking only the non-trivial Π case. If for each $x \in A$, $E_{B[x]}$ is transitive then:

$$E_{\Pi x \in A.B[x]}(x,y) \;\&\; E_{\Pi x \in A.B[x]}(y,z) \;\&\; \xi',\zeta' \in Obj'(|A|') \;\&\; E_A(p(\xi'),p(\zeta'))$$

$$\Rightarrow \; E_A(p(\xi'),p(\zeta'))$$

$$\Rightarrow \; E_{B[\underline{\xi}']}(x|p(\xi'),y|p(\xi')) \;\&\; E_{B[\underline{\xi}']}(y|p(\xi'),z|p(\zeta'))$$

$$\Rightarrow \; E_{B[\underline{\xi}']}(x|p(\xi'),z|p(\zeta')) \qquad \text{by hypothesis}$$

Now, define another relation $I_A \in D \times D \to V_n$ for each $A \in V_n$, by induction over the typestructure, in the same way as E_A. Take $I_{V_n} =_{df} E_{V_n}$, $I_N =_{df} E_N$, $I_{\Sigma x \in A.B[x]} =_{df} E_{\Sigma x \in A.B[x]}$ and $I_{a=b} =_{df} E_{a=b}$; but define $I_{\Pi x \in A.B[x]}$ as follows:

$$I_{\Pi x \in A.B[x]}(x,y) =_{df} \Pi \xi' \in Obj'(|A|').I_{B[\underline{\xi}']}(x|p(\xi'),y|p(\xi'))$$

Then observe that, $\Pi \xi',\zeta' \in Obj'(|C|').E_C(p(\xi'),p(\zeta')) \leftrightarrow I_C(p(\xi'),p(\zeta'))$. Again show this by induction and there is just the Π case to check. Suppose that for each $x \in A$, it holds of $B[x]$ then:

$$\chi_1',\chi_2' \in Obj'(|\Pi x \in A.B[x]|') \;\&\; E_{\Pi x \in A.B[x]}(p(\chi_1'),p(\chi_2')) \;\&\; \xi' \in Obj'(|A|')$$

$$\Rightarrow E_A(p(\xi'),p(\xi'))$$

$$\Rightarrow E_{B[\underline{\xi}']}(p(\chi_1')|p(\xi'),p(\chi_2')|p(\xi'))$$

$$\Rightarrow I_{B[\underline{\xi}']}(p(\chi_1')|p(\xi'),p(\chi_2')|p(\xi')) \qquad \text{by hypothesis}$$

hence $I_{\Pi x \in A.B[x]}(p(\chi_1'),p(\chi_2'))$.

$$\chi_1',\chi_2' \in Obj'(|\Pi x \in A.B[x]|') \;\&\; I_{\Pi x \in A.B[x]}(p(\chi_1'),p(\chi_2')) \;\&\; \xi',\zeta' \in Obj'(|A|') \;\&\; E_A(p(\xi'),p(\zeta'))$$

$$\Rightarrow E_{\Pi x \in A.B[x]}(p(\chi_2'),p(\chi_2'))$$

$$\Rightarrow E_{B[\underline{\xi}']}(p(\chi_1')|p(\xi'),p(\chi_2')|p(\xi')) \;\&\; E_{B[\underline{\ }']}(p(\chi_2')|p(\xi'),p(\chi_2')|p(\ '))$$

$$\Rightarrow E_{B[\underline{\xi}']}(p(\chi_1')|p(\xi'),p(\chi_2')|p(\zeta')) \qquad \text{by transitivity}$$

hence $E_{\Pi x \in A.B[x]}(p(\chi_1'),p(\chi_2'))$.

3.3.1.16. ext(n) is verified in an extensional submodel

We begin by showing by induction that for $\chi_1',\chi_2' \in Obj'(|m|')$

$E_m(p(\chi_1'),p(\chi_2')) \leftrightarrow Obj'(|\chi_1' \underline{=}_{ext} \chi_2'|')$.

If $\chi_1',\chi_2' \in Obj'(|N|')$ then from 2.4.1.(1) we have:

$p(\chi_1') \simeq p(\chi_2') \leftrightarrow Obj'(|\chi_1' = \chi_2'|')$.

If $\chi_1',\chi_2' \in Obj'(|m+1|')$ then we have:

$$E_{m+1}(p(\chi_1'),p(\chi_2')) \Rightarrow I_{m+1}(p(\chi_1'),p(\chi_2'))$$

$$\Rightarrow \Pi\xi' \in Obj'(|m|').E_N(p(\chi_1')|p(\xi'),p(\chi_2')|p(\xi'))$$

$$\Rightarrow \Pi\xi' \in Obj'(|m|').Obj'(|\chi_1'(\underline{\xi}') = \chi_2'(\underline{\xi}')|')$$

$$\Rightarrow Obj'(|\underline{\chi_1'} \underline{=}_{ext} \underline{\chi_2'}|')$$

The last inference is made by taking $\Lambda\alpha.p(\chi_1')|\alpha$ and observing that $\Psi_{\underline{\chi_1'} \underline{=}_{ext} \underline{\chi_2'}}(\Lambda\alpha.p(\chi_1')|\alpha)$ as $E_{\underline{\chi_1'}(\underline{\xi}')=\underline{\chi_2'}(\underline{\xi}')}(x,y) =_{df} T$.

We can now show that ext(n) is verified. Proceeding by induction, ext(N) is trivially verified, and for $\chi' \in Obj'(|m+1|')$:

$$\Psi_{m+1}(p(\chi')) =_{df} \Pi\xi',\zeta' \in Obj'(|m|').E_m(p(\xi'),p(\zeta')) \rightarrow E_N(p(\chi')|p(\xi'),p(\chi')|p(\zeta').$$

$$\Leftrightarrow \Pi\xi',\zeta' \in Obj'(|m|').Obj'(|\xi' \underline{=}_{ext} \zeta'|') \rightarrow Obj'(|\chi'(\underline{\xi}') = \chi'(\underline{\zeta}')|').$$

hence $Obj'(|ext(m+1)|')$.

The last inference is made by taking $\Lambda\alpha.\Lambda\beta.\gamma.\Lambda\delta.\alpha|\beta$ and again observing that $\Psi_{ext(m+1)}(\Lambda\alpha.\Lambda\beta.\gamma.\Lambda\delta.\alpha|\beta)$ because, trivially, $E_{\chi'(\underline{\xi}')=\chi'(\underline{\zeta}')}(x,y) =_{df} T$.

3.3.1.17. Theorem. ext(n) is consistent with M-L, for each n.

3.3.2. Compactness

Another useful heridary property is compactness or boundedness on closed intervals. An example of the use of this property is to be found in Howard [1973], appendix.

Firstly define an order relation over the higher types as follows:

if $n,m \in N$ then $n \leq m =_{df} \Sigma r \in N.m=n+r$

if $f,g \in A \to B$ then $f \leq g =_{df} \Pi x \in A.f(x) \leq g(x)$

A function is compact if there is a function which majorizes it with respect to this ordering. So we define axioms of compactness for the higher types as follows:

$comp(N) =_{df} T$

$comp(A \to B) =_{df} \Pi F \in A \to B.\Sigma G \in A \to B.\Pi f,g \in A.f \leq g \to F(f) \leq G(g).$

3.3.2.1. The compact submodel of an r.s.

We proceed in a similar way to 3.3.1.14. Define for each $A \in V_n$, C_A and $M_A \in D \times D \to V_n$ and then produce the submodel by:

i) $Typ'_k =_{df} D \times (D \to V_n) \times (D \times (D \to V_n) \to V_n)$

ii) for each $\phi \in Typ'_k$:

$$Obj'(\phi) =_{df} \Sigma x \in D.p(q(\phi))(x) \;\&\; q(q(\phi))(x,p(q(\phi))).$$

iii) the definition of $|.|'$ is such that:

$$q(q(|A|')) =_{df} \Psi_A$$

where $\Psi_A =_{df} \lambda x.\lambda f.\Sigma y \in D.f(y) \;\&\; C_A(y,y) \;\&\; C_A(x,y)$

The motivation behind the definition of C and M being that:

$$M_n(p(\xi'),p(\zeta')) \Rightarrow Obj'(|\underline{\xi}' \leq \underline{\zeta}'|')$$

and $\Pi \chi' \in Obj'(|n|').\Psi_n(p(\chi')). \Rightarrow Obj'(|comp(n)|')$

As before we extract the clause of the definition of C and M from that of |.|': and in addition follow the proceedure of page 63 & 63(a)

$C_{V_n}(x,y) =_{df} D \to V_n$

$C_{\Pi x \in A.B[x]}(x,y) =_{df} \Pi \xi', \zeta' \in Obj'(|A|').M_A(p(\xi'),p(\zeta')) \to M_{B[\underline{\xi}']}(x|p(\xi'),y|p(\zeta'))$.

$C_N(x,y) =_{df} \Sigma n \in N.y \simeq \$^n(x)$

$C_{\Sigma x \in A.B[x]}(x,y) =_{df} T$

$C_{a=b}(x,y) =_{df} T$

$M_{V_n} =_{df} C_{V_n}, M_N =_{df} C_N, M_{\Sigma x \in A.B[x]} =_{df} C_{\Sigma x \in A.B[x]}$ and $M_{a=b} =_{df} C_{a=b}$ but

$M_{\Pi x \in A.B[x]}$ is defined by:

$M_{\Pi x \in A.B[x]}(x,y) =_{df} \Pi \xi' \in Obj'(|A|').C_{B[\underline{\xi}']}(x|p(\xi'),y|p(\xi'))$.

To complete the definition of |.|' the induction hypothesis must be strengthened. If for $x_1 \in A_1$, $x_2 \in A_2[x_1],\ldots,x_n \in A_n[x_1,\ldots,x_{n-1}]$, $b[x_1,\ldots,x_n] \in B[x_1,\ldots,x_n]$ then in addition we require:

for $\xi_1' \in Obj'(|A_1|'),\ldots\ldots,\xi_n' \in Obj'(|A_n[\xi_1',\ldots,\xi_{n-1}']|')$

$|b[\xi_1',\ldots,\xi_n']|$ and $\bar{b}[\xi_1',\ldots,\xi_n'] \in Obj'(|B[\xi_1',\ldots,\xi_n']|')$

$\|b[\xi_1',\ldots,\xi_n']\|$ and $\bar{\bar{b}} \in S_n$

in such a way that:

$\|b[\xi_1',\ldots,\xi_n']\|\ (p(\xi_1'),p(\xi_2'),\ldots,p(\xi_n')) \simeq p(|b[\xi_1',\ldots,\xi_n']|')$

and $\bar{\bar{b}}(p(\xi_1'),p(\xi_n'),\ldots,p(\xi_n')) \simeq p(\bar{b}[\xi_1',\ldots,\xi_n'])$

And if $\zeta_1' \in Obj'(|A_1|'),\ldots\ldots,\zeta_n' \in Obj'(|A_n[\zeta_1',\ldots,\zeta_{n-1}']|')$

$$C_{A_1}(p(\xi_1'),p(\zeta_1')) \;\&\; \ldots\ldots \;\&\; C_{A_n[\underline{\xi}_1',\ldots,\underline{\xi}_{n-1}']}(p(\xi_n'),p(\zeta_n')). \rightarrow$$

$$C_{B[\underline{\xi}_1',\ldots,\underline{\xi}_n']}(p(b[\xi_1',\ldots,\xi_n']),p(b[\zeta_1',\ldots,\zeta_n']))$$

$$\&\; C_{B[\underline{\xi}_1',\ldots,\underline{\xi}_n']}(p(|b[\xi_1',\ldots,\xi_n']|'),p(b[\xi_1',\ldots,\xi_n'])).$$

and then, for example, we have:

$$C_{\Pi(A[\underline{\xi}'],B[\underline{\xi}'x])}(p(|\lambda b[\xi_1',x]|'),\bar{\bar{b}})$$

3.3.2.2. comp(n) is verified in a compact submodel

Similarly to 3.3.1.16. begin by showing that for $\chi_1',\chi_2' \in Obj'(|m|')$ $M_m(p\chi_1'),p(\chi_2')) \leftrightarrow Obj'(|\underline{\chi}_1' \leq \underline{\chi}_2'|')$. This goes through in the same way as 3.3.1.16. if we observe that $M_{\Sigma r \in N.\underline{\chi}_2'=\underline{\chi}_1'+r}(x,y) =_{df} T$

Then show by induction that comp(n) is verified. For $\chi_1' \in Obj'(|m+1|')$:

$$\Psi_{m+1}(p(\chi_1')) =_{df} \Sigma\chi_2' \in Obj'(|m+1|').\Pi\xi',\zeta' \in Obj'(|m|').M_m(p(\xi'),p(\zeta'))$$

$$\rightarrow M_N(p(\chi_1')|p(\xi'),p(\chi_2')|p(\zeta')).$$

$$\Leftrightarrow \quad \Sigma\chi_2' \in Obj'(|m+1|').\Pi\xi',\zeta' \in Obj'(|m|').Obj'(|\underline{\xi}' \underline{\leq} \underline{\zeta}'|')$$

$$\rightarrow Obj'(|\underline{\chi}_1'(\underline{\xi}') \underline{\leq} \underline{\chi}_2'(\underline{\zeta}')|')$$

hence Obj'(|comp(m+1)|'). Where the object of this type is obtained by taking $\Lambda\alpha.[\alpha',\Lambda\beta.\gamma.\Lambda\delta.[\alpha'|\beta-\alpha|\beta,\alpha|\beta]]$ where $\alpha'=p(p(\Psi_{m+1}(\alpha)))$ and observing that $\Psi_{comp(m)}(\Lambda\alpha.[\alpha',\Lambda\beta.\gamma.\Lambda\delta.[\alpha'|\beta-\alpha|\beta,\alpha|\beta]])$ as again, trivially, $C_{\Sigma G \in m+1.etc}(x,y) =_{df} T$.

3.3.2.3. Theorem. comp(n) is consistent with M-L, for each n.

3.3.2.4. The compact extentional submodel

This is constructed by combining the extentional and compact submodels. So taking Typ' and Obj' to be defined as in the compact submodel:

$\Psi_A =_{df} \lambda x.\lambda f.\ E_A(x,x)\ \&\ \Sigma y \in D.f(y)\ \&\ C_A(y,y)\ \&\ C_A(x,y).$

The definition of $|.|'$ proceeds as before and we combine the induction hypotheses of the extentional and compact submodels.

3.3.2.5. Theorem. ext(n) and comp(n) are mutually consistent, for each n.

To show this combine the arguments of 3.3.1.16. and 3.3.2.2.

§4. Generalizations of results over the type structure.

The complex type structure of M-L allows us to generalize the definitions of such notions as continuity, recursiveness, extensionality and compactness more than is usual. That is, we can define predicates of every type in such a way that at a finite type or an arithmetic type the predicate expresses the usual property at that type. The difficulty is to make the generalization sensible.

3.4.1. Continuity

Troelstra has shown that extending type two continuity even up to type three, in a natural way, leads to contradiction. (See Troelstra [1973], page 159.) The extension is:

$$\text{cont}(3) =_{df} \Pi A \in ((N\to N)\to N)\to N.\forall F^2.\exists f_1,\ldots,f_k.\forall G^2.$$

$$F(f_1)=G(f_1)\ \&\ \ldots\ldots\ \&\ F(f_k)=G(f_k) \to A(F)=A(G)$$

And we can show that $\vdash\sim\text{cont}(3)$.

Another approach, which is similar to that used by Martin-Löf and Hancock in connection with their topological model of Martin-Löf type theory (unpublished work), is to define what it is for an object of each type to have an associate, the associated object being a member of a fixed type. We shall use the type of effectively enumerable sets; N→N+T. Define a relation 'as' over the type structure in such a way that for $A \in V_n$; $as_A \in A\times(N\to N+T) \to V_n$: following the proceedure of page 63:

if $Z \in V_n$ then $as_{V_n}(Z,X) =_{df} T$

if $f \in \Pi x \in A.B[x]$ then $as_{\Pi x \in A.B[x]}(f,X) =_{df} \Pi x \in A.\Pi Y \in (N \to N+T).$

$$as_A(x,Y) \to {}^{\Sigma} Z \in (N \to N+T).Z \simeq X|Y \;\&\; as_{B[x]}(f(x),Z).$$

if $n \in N$ then $as_N(n,X) =_{df} \{\hat{n}\} \simeq X$

if $y \in \Sigma x \in A.B[x]$ then $as_{\Sigma x \in A.B[x]}(y,X) =_{df} \Sigma Z_1, Z_2 \in (N \to N+T).Z_1 \simeq p(X) \;\&\; Z_2 \simeq q(X)$

$$\& \; as_A(p(y),Z_1) \;\&\; as_{B[p(y)]}(q(y),Z)$$

if $z \in a=b$ then $as_{a=b}(z,X) =_{df} T$

where '$\simeq$','$|$','p' and 'q' are as in 2.4.3.

Then for any type C we can define continuity at C by:

$g.cont(C) =_{df} \Pi x \in C.\Sigma X \in (N \to N+T).\ as_C(x,X)$

and clearly $\vdash g.cont(2) \to s.cont(2)$, because associating an effectively enumerable set to an object of type two is more general than associating a number theoretic function.

3.4.1.1. Theorem. g.cont(C) is consistent with M-L, for each C

Use the set realizability model, and show by induction over the type-structure that for each C:

$$\Pi \xi \in Obj(|C|).Obj(|as_C(\underline{\xi},p(\xi))|)$$

so that the object which realizes $as_C(\underline{\xi},p(\xi))$ is continuous in $p(\xi)$. The Π case is the crucial one to check, and it follows from the fact that $p(|\underline{\zeta}(\xi)|) \simeq p(\zeta)|p(\xi)$.

Then taking $\chi = |p(\xi)|'$ we are able to show that:

$$\Pi \xi \in Obj(|C|).\Sigma \chi \in Obj(|N \to N+T|).Obj(|as_C(\underline{\xi},\underline{\chi})|)$$

and the inductively proved continuity condition gives us an object of type Obj(|g.cont(C)|) by Λ-abstraction.

3.4.1.2. Remark

The same procedure will lead to a generalization of Church's Thesis if we define what it is to be a recursive associate of an object at each type.

3.4.2. Extensionality

Begin by defining an order relation by induction over the typestructure following the proceedure of page 65;

if $X,Y \in V_n$ then $X \underset{ext}{=} Y =_{df} \Sigma F \in X \to Y.\Sigma G \in Y \to X.\Pi x \in X.G(F(x))=x$

$$\& \Pi y \in Y.F(G(y))=y \;\&\; \Pi x_1, x_2 \in X.\; x_1 \underset{ext}{=} x_2 \to F(x_1) \underset{ext}{=} F(x_2)$$

$$\& \Pi y_1, y_2 \in Y.\; y_1 \underset{ext}{=} y_2 \to G(y_1) \underset{ext}{=} G(y_2).$$

if $f,g \in \Pi x \in A.B[x]$ then $f \underset{ext}{=} g =_{df} \Pi x \in A.f(x) \underset{ext}{=} g(x)$

if $n,m \in N$ then $n \underset{ext}{=} m =_{df} n=m$

if $x,y \in \Sigma x \in A.B[x]$ then $x \underset{ext}{=} y =_{df} p(x) \underset{ext}{=} p(y)$ & $\Sigma X \in B[p(x)] \underset{ext}{=} B[p(y)]$.

$$p(X)(q(x)) \underset{ext}{=} q(y).$$

If $z,t \in a=b$ then $z \underset{ext}{=} t =_{df} T$.

And then define the axioms over the type structure:

$g.ext(N) = g.ext(V_n) = g.ext(a=b) =_{df} T$ and

$g.ext(\Pi x \in A.B[x]) =_{df} \Pi f \in \Pi x \in A.B[x].\Pi x,y \in A.x \underset{ext}{=} y \to \Sigma X \in B[x] \underset{ext}{=} B[y]$.

$$p(X)(f(x)) \underset{ext}{=} f(y)$$

$g.ext(\Sigma x \in A.B[x]) =_{df} g.ext(A) \;\&\; \Pi x \in A.g.ext(B[x])$

Then in fact, $\vdash g.ext(A \to B) \leftrightarrow ext(A \to B)$, which is shown by taking an $X \in B \to B$

such that $p(X) =_{ext} \lambda x.x$. And a similar procedure to generalize compactness could be undertaken.

§5. Stronger continuity principles

In this section we shall develop the formal counterparts of four stronger continuity principles and establish the consistency of two of them.

3.5.1. Bounded uniform continuity

At type two this is the principle that a type two object is uniformly continuous, if its domain is restricted to a bounded interval of Baire Space. Formally we define:

$$\text{b.u.cont}(2) =_{df} \forall F^2.\forall f.\exists m.\forall h{\leq}f.\forall g.g \in \bar{h}(m) \to F(h)=F(g).$$

where '$\forall h{\leq}f$' is an abbreviation for '$\Pi h \in N\to N.\Pi x \in N.h(x){\leq}f(x)$'.

And we have the following:

3.5.1.1. $\vdash$ comp(2) & cont(2) → b.u.cont(2)

Let Φ_F be the modulus of continuity functional obtained from cont(2) by a.c.(1,0) for a fixed $F \in (N\to N)\to N$. Then by comp(2), Φ_F is compact and let Φ'_F be its majorizing functional. So for $f \in N\to N$, taking $m =_{df} \Phi'_F(f)$:

$$\Phi_F(h) \leq m$$

$$g \in \bar{h}(m) \to g \in \bar{h}(\Phi'_F(h))$$

and $$g \in \bar{h}(\Phi_F(h)) \to F(h)=F(g)$$

hence result.

3.5.1.2. $\vdash$ b.u.cont(2) → cont(2)

Just take h=f.

Classically, compact objects of type two are not necessarily continuous: for example the function defined by the (classical) schema:

$$F(f) =_{df} \begin{cases} 0 & \text{if } f =_{ext} \lambda x.0 \\ 1 & \text{o.w.} \end{cases}$$

is clearly compact, but is discontinuous (no finite amount of information about f will tell me if it is extensionally equal to $\lambda x.0$). Consequently, we have $\vdash$ (comp(2) & cont(2) & dec(2)), as the above function, F, would be definable in M-L. But in fact:

3.5.1.3. Theorem. comp(n) and g.cont(n) are mutually consistent, for each n.

To show this repeat the argument of 3.4.1.1. using the compact submodel of the set realizability model. This goes through in the same way, except for the last step where the Λ-abstracted object must be proved to be majorized, but as in previous cases this is trivially true because $\Psi_{\Sigma x \in N \to N+T.etc}(x,y) =_{df} T$.

Hence in the submodel both comp(2) and cont(2) are validated, and the result follows.

3.5.1.4. Corollary. b.u.cont(2) is consistent with M-L.

3.5.2. The fan theorem

It is usual to note that a fan is determined by a number-theoretic function f, so that g is in the fan if $g \leq f$; or for us if $\Sigma g \in N \to N.g<f$. But it can be shown (c.f. Troelstra [1973] pp. 80/1) that the fan theorem stated for an arbitrary fan is equivalent to that for the binary spread (by coding); and we can make use of the expressive power of M-L to define the binary spread as the type: $N \to N_2$. Hence we define:

$$fan_R =_{df} \forall f \in N \to N_2.\exists m.R(f,m) \to \exists \acute{n}.\forall f \in N \to N_2.\exists m.\forall g \in N \to N_2.f \in \bar{g}(n) \to R(g,m)$$

and again we define the axiom:

$$\text{fan} =_{df} \forall F \in (N\to N_2)\to N.\exists n.\forall f,g \in N\to N_2.f \in \bar{g}(n) \to F(f)=F(g).$$

Clearly $\vdash$ fan $\to$ ext$((N\to N_2)\to N)$ and comparing b.u.cont(2) and fan it can be seen that:

3.5.2.1. $\vdash$ b.u.cont(2) & g.ext$((N\to N_2)\to N) \to$ fan.

Given $F \in (N\to N_2)\to N$ define $F' \in 2$ by:

$$F'(\alpha) =_{df} F(\alpha^*) \qquad \text{where}$$

$$\alpha^*(n) =_{df} \begin{cases} 1^* & \alpha(n)=0 \\ 2^* & \alpha(n)=1 \\ 1^* & \text{o.w.} \end{cases}$$

and for $f \in N\to N_2$ define $f' \in N\to N$ by:

$$f'(n) =_{df} \begin{cases} 0 & \text{if } f(n)=1^* \\ 1 & \text{if } f(n)=2^* \end{cases}$$

then $(f')^* \underset{ext}{=} f$ and

If $F \in (N\to N_2)\to N$; $F' \in 2$, $\lambda x.1 \in N\to N$, and b.u.cont(2) gives an $n \in N$ such that for all $f,g \in N\to N_2$:

$$f' \leq \lambda x.1$$

and so: $g' \in \overline{f'}(n) \to F'(f') = F'(g')$

$$F((f')^*) = F((g')^*)$$

$$F(f) = F(g) \text{ by ext}((N\to N_2)\to N)$$

and $g \in \bar{f}(n) \leftrightarrow g' \in \overline{f'}(n)$; hence result.

3.5.2.2. Theorem. fan is consistent with M-L.

Use the compact extensional submodel of the set realizability model, and repeating the argument of 3.5.1.3. we have that cont(2), comp(2) and ext((N→N_2)→N) are validated there. In fact, ext((N→N_2)→N) is validated in any extentional submodel: the proof is just that of 3.3.1.16. taking account of the fact that $E_{N_2}(x,y) =_{df} T$. The result follows from 3.5.1.1. and 3.5.2.1.

3.5.3. Inductively defined continuity

As yet we have not made use of inductive definitions to extend M-L and allow a stronger definition of continuity. Such a suggestion has been made by Peter Hancock (private communication). Firstly we define what it is for a natural, c number to immediately secure a type two functional, F:

$$i.s.(F,c) =_{df} \forall f,g.\forall m \leq lh(c).f(m)=g(m)=c_m \rightarrow F(f)=F(g).$$

and then give an inductive definition of 'c secures F':

$$\frac{s \in i.s.(F,c)}{\rho(x) \in sec(F,c)} \qquad \frac{f \in \Pi x \in N(sec(F,c*x)}{\cap f \in sec(F,c)}$$

which generates the schema of induction on the prcof of 'c secures F':

$$\frac{t \in sec(F,c) \qquad \begin{array}{c} u \in i.s.(F,c) \\ \vdots \\ a[u] \in P[\rho(u),F,c] \end{array} \qquad \begin{array}{c} f \in \Pi x \in N.sec(F,c*x) \\ g \in \Pi x \in N.P[f(x),F,c*x] \\ \hline b[f,g] \in P\ [\ f,F,c] \end{array}}{S_{a,b}[t] \in P[t,F,c]}$$

That is, more formally, we add new function constants ρ, $\cap$, and S to the language of M-L and the function constants are introduced in the appropriate way.

We can now state the inductively defined axiom of type two continuity, as follows:

$$i.d.cont(2) =_{df} \forall F^2.sec(F,<>)$$

Furthermore, cont(2) can be expressed using these newly defined notions, by:

$$\mathrm{cont}(2) =_{df} \forall F.\forall f.\exists n.\mathrm{i.s.}(F,\bar{f}(n))$$

And we have the following:

3.5.3.1. $\vdash \mathrm{i.d.cont}(2) \to \mathrm{cont}(2)$

We prove this by induction on a proof of sec(F,c); assume that $t \in \mathrm{sec}(F,c)$:

$$\frac{u \in \mathrm{i.s.}(F,c)}{\lambda f.(\mathrm{lh}(u),u) \in \forall f \in c.\exists n.\mathrm{i.s.}(F,\bar{f}(n))}$$

and:

$$\frac{g \in \Pi x \in N.\forall f \in c^{*}x.\exists n.\mathrm{i.s.}(F,\bar{f}(n))}{\lambda f.g(f(\mathrm{lh}(c)+1))(f) \in \forall f \in c.\exists n.\mathrm{i.s.}(F,\bar{f}(n))}$$

then:

$$S[t] \in \forall f \in c.\exists n.\mathrm{i.s.}(F,\bar{f}(n))$$

so we have:

$$\forall F^{2}.\forall c.\mathrm{sec}(F,c) \to \forall f \in c.\exists n.\mathrm{i.s.}(F,\bar{f}(n)).$$

And in particular:

$$\forall F^{2}.\mathrm{sec}(F,\langle\rangle) \to \forall f.\exists n.\mathrm{i.s.}(F,\bar{f}(n)).$$

3.5.3.2. Classically, cont(2) and i.d.cont(2) are equivalent

$$\begin{aligned}
\sim\mathrm{sec}(F,\langle\rangle) &\Rightarrow \sim\forall x.\mathrm{sec}(F,\langle x\rangle) \\
&\Rightarrow \exists x.\sim\mathrm{sec}(F,\langle x\rangle) && \text{- classically} \\
&\Rightarrow \exists x,y.\sim\mathrm{sec}(F,\langle x,y\rangle) && \text{- classically} \\
&\;\;\vdots \\
&\Rightarrow \exists f.\forall n.\sim\mathrm{sec}(F,\bar{f}(n)) && \text{- by repetition} \\
&\Rightarrow \sim\forall f.\exists n.\mathrm{sec}(F,\bar{f}(n))
\end{aligned}$$

3.5.4. Bar induction

One formal counterpart of bar induction is the following schema, which is just Kleene's axiom *26.3c (Kleene-Vesley [1965], p. 55):

$$b.i. = (\forall f.\exists n.P(\bar{f}(n)).\ \&\ \forall c.(P(c) \to Q(c))\ \&\ \forall c.(\forall x.Q(c*x) \to Q(c))) \to Q(<>).$$

In b.i. no restrictions are put on the predicate P, but we need the version with a monotonicity condition:

$$b.i.(mon) =_{df} \text{the antecedants of b.i. } \&\ (\forall c,x.P(c) \to P(c*x).) \to Q(<>).$$

We now show that i.d.cont(2) can be 'reached' from cont(2) by using b.i.(mon) and that i.d.cont(2) is stronger than b.i.(mon):

3.5.4.1. $\vdash$ cont(2) & b.i.(mon) $\to$ i.d.cont(2)

Fix F and take Q(c) to be sec(F,c) and P(c) to be i.s.(F,c) then the antecedants are just cont(2), the clauses of the inductive definition of 'secured' and the obvious proof of the monotonicity of 'immediately secured'. The conclusion is i.d.cont(2).

3.5.4.2. $\vdash$ i.d.cont(2) $\to$ b.i.(mon)

Suppose the antecedant of b.i.(mon) and in particular use the axiom of choice on $\forall f.\exists n.P(\bar{f}(n))$ to get $\exists F^2.\forall f.P(\bar{f}(F(f)))$. Then apply i.d.cont(2) to that F and prove Q(c) by induction over the proof that sec(F,c), as follows:

Assuming that i.s.(F,c) then consider, for each m with $lh(m) = F([c]) - lh(c)$, the function $[c*m]$.

By assumption we have:

$$F([c]) = F([c*m])$$

and $$P(\overline{[c*m]}(F([c*m]))\ )$$

so $$P(\overline{[c*m]}(F([c]))\ )$$

Then if $F([c]) < lh(c)$ then by the assumed monotonicity of P:

$$P(c*m)$$

and if F([c]) > lh(c) then lh(c*m)=F([c]) and so:

$$P(c*m)$$

Hence either way we have Q(c*m) and

$$Q(c*\langle m_o,\ldots,m_{lh(m)\dot{-}2}\rangle*x)$$

and so by assumption

$$Q(c*\langle m_o,\ldots,m_{lh(m)\dot{-}2}\rangle)$$

Then repeat the argument for each n with lh(n)=F([c])-(lh(c)+1) to get Q(c*n). And eventually Q(c).

Completing the induction:

$$\dfrac{\begin{array}{ll} & \\ \vdots & \dfrac{\forall x.Q(c*x)}{Q(c)} \\ \mathrm{sec}(F,c) & Q(c) \end{array}}{Q(c)}\ \mathrm{i.s.}(F,c)$$

Chapter 4

OTHER THEORIES FORMALIZED

It has been claimed that Martin-Löf type theory is a complete theory of constructions; that in fact all the constructions of Bishop [1967] can be made inside it. It is already apparent that the theory has a high expressive power. We demonstrate these properties by indicating how to formalize the constructive theory of reals, an intuitionistic set theory and all of category theory inside M-L.

§1. The theory of real numbers

We begin by defining a type Q of rational numbers:

$$Q =_{df} \Sigma x \in N_2 \times N \times N.\, q(q(x)) \neq 0$$

A rational number is a triple comprising a sign (1* for positive, 2* for negative), a numerator and denominator, together with a proof that the denominator is non-zero. And we could adopt the abbreviation $\frac{x_2}{x_3}$ for $((x_1,(x_2,x_3)),f) \in Q$.

Define the ordering $<_Q$ and an identity $\equiv_Q$ on Q; and also the modulus $|.|_Q$ and negation $-(.)$ of a member of Q:

$$\begin{aligned}((x_1,(x_2,x_3)),f) <_Q ((y_1,(y_2,y_3)),g) =_{df}\ & x_1=y_1=1^* \ \&\ x_2 \cdot y_3 < y_2 \cdot x_3 \\ & + x_1=y_1=2^* \ \&\ y_2 \cdot x_3 < x_2 \cdot y_3 \\ & + x_1=2^* \ \&\ y_1=1^*\end{aligned}$$

$$\begin{aligned}((x_1,(x_2,x_3)),f) \equiv_Q ((y_1,(y_2,y_3)),g) =_{df}\ & \Sigma n \in N.\, n \cdot x_2 = y_2 \ \&\ n \cdot x_3 = y_3. \\ & + \Sigma m \in N.\, m \cdot y_2 = x_2 \ \&\ m \cdot y_2 = x_2.\end{aligned}$$

$$|((x_1,(x_2,x_3)),f)|_Q =_{df} ((1^*,(x_2,x_3)),f)$$

$$-((x_1,(x_2,x_3)),f) =_{df} \begin{cases} ((1^*,(x_2,x_3)),f) & \text{if } x_1 = 2^* \\ ((2^*,(x_2,x_3)),f) & \text{if } x_1 = 1^* \end{cases}$$

Now, a countable sequence of rational numbers is an object of type $N \to Q$, and we say that a countable sequence of rationals, f, is cauchy if cau(f), where:

$$cau(f) =_{df} \Pi n \in N.\Sigma m \in N.\Pi p \in N.\,|f(m)-f(m+p)|_Q <_Q \frac{1}{n+1}$$

Then we are in a position to define the type R of real numbers by:

$$R =_{df} \Sigma f \in N \to Q.cau(f)$$

And define an ordering $<_R$ and an identity $\equiv_R$ on R by:

$$(f,p_1) <_R (g,p_2) =_{df} \Sigma n,k \in N.\Pi p \in N.\,g(n+p)-f(n+p)_Q > \frac{1}{k+1}$$

$$(f,p_1) \equiv_R (g,p_2) =_{df} \Pi n \in N.\Sigma m \in N.\Pi p \in N.\,|f(m+p)-g(m+p)|_Q <_Q \frac{1}{n+1}$$

The operations $|\ |_R$, $-(\ )$ and $(\ )+(\ )$ on R are defined in the obvious way.

4.1.1. Continuity on $R \to R$.

Clearly, not all objects of type $R \to R$ are extentional with respect to the identity relation $\equiv_R$, and so they are not all continuous. For example, take $F \in R \to R$ defined by:

$$F((f,p)) =_{df} (\lambda x.f(0),\ \text{proof that } cau(\lambda x.f(0)))$$

and two objects $(f_1,p_1),(f_2,p_2) \in R$, such that $f_1(0) \neq f_2(0)$ and so $F((f_1,p_1)) \not\equiv_R F((f_2,p_2))$ but $(f_1,p_1) \equiv_R (f_2,p_2)$.

In defining continuity, then, we must restrict ourselves to maps with respect to the relation $\equiv_R$. The type of maps from R to R, M(R,R), can then be defined by:

$$M(R,R) =_{df} \Sigma F \in R \to R.\Pi r,s \in R.\, r \equiv_R s \to F(r) \equiv_R F(s).$$

And the axiom stating that every map from R to R is continuous is defined by:

$$cont(R \to R) =_{df} \Pi F \in M(R,R).\Pi r \in R.\Pi m \in N.\Sigma n \in N.\Pi s \in R.$$

$$|r-s| < \frac{1}{2^{(n+2)}} \to |F(r)-F(s)| < \frac{1}{m+1}$$

4.1.2. The types $R \to R$ and $1 \to 1$

Coding Q into N, we can think of R as a subtype of $N \to N$. This prompts us to define a metric d on $N \to N$, as follows:

for $f,g \in N \to N$, $d(f,g) =_{df} (d_1(f,g),d_2(f,g)) \in R$ where

$$d_1(f,g)(s(x)) =_{df} \begin{cases} \frac{1}{s(s(x))} & \text{if } f \in \bar{g}(x) \\ d_1(f,g)(x) & \text{o.w.} \end{cases}$$

$$d_1(f,g)(0) =_{df} 1$$

$d_2(f,g)$ is the proof that $cau(d_1(f,g))$; that is, a formalization of:

for $x \in N$, if $f \in \bar{g}(x)$ then $d_1(f,g)(x) < \frac{1}{x+1}$ and for $y \in N$

$d_1(f,g)(x) \geq d_1(f,g)(x+y)$, so $d_1(f,g)(x)-d_1(f,g)(x+y) \leq d_1(f,g)(x) < \frac{1}{x+1}$.

if $f \notin \bar{g}(x)$ then for $y \in N$, $d_1(f,g)(x)=d_1(f,g)(x+y)$ so

$d_1(f,g)(x)-d_1(f,g)(x+y) = 0 < \frac{1}{x+1}$.

We can now state an axiom of continuity at type $1\to1$ using this metric, which is analogous to $cont(R \to R)$:

$$cont_{\varepsilon,\delta}(1\to1) =_{df} \forall F^{1\to1}.\forall n.\forall f.\exists m.\forall g.d_1(f,g) < \frac{1}{m+1} \to d_1(F(f),F(g)) < \frac{1}{n+1}.$$

and can prove that $\vdash cont\ (1\to1) \leftrightarrow cont_{\varepsilon,\delta}(1\to1)$. This follows from the observation that $d_1(f,g) < \frac{1}{x+1}$ iff $f \in \bar{g}(x)$.

4.1.3. $\vdash cont(1\to1) \to cont(R\to R)$

The proof of this would be a formalization of the similar theorem of Heiting [1956] p. 46. We only need define what it is for a real to be canonical:

$$can((f,p)) =_{df} \Pi n \in N.\Sigma m \in N.\ m=f(n)\cdot 2^n \ \&\ f(n)-f(n+1) \leq \frac{1}{2^{(n+1)}}$$

and formalize the proof of Heiting [1956] p. 42 that:

$$\Pi r \in R.\Sigma s \in R.\ r \equiv_R s \ \&\ can(s).$$

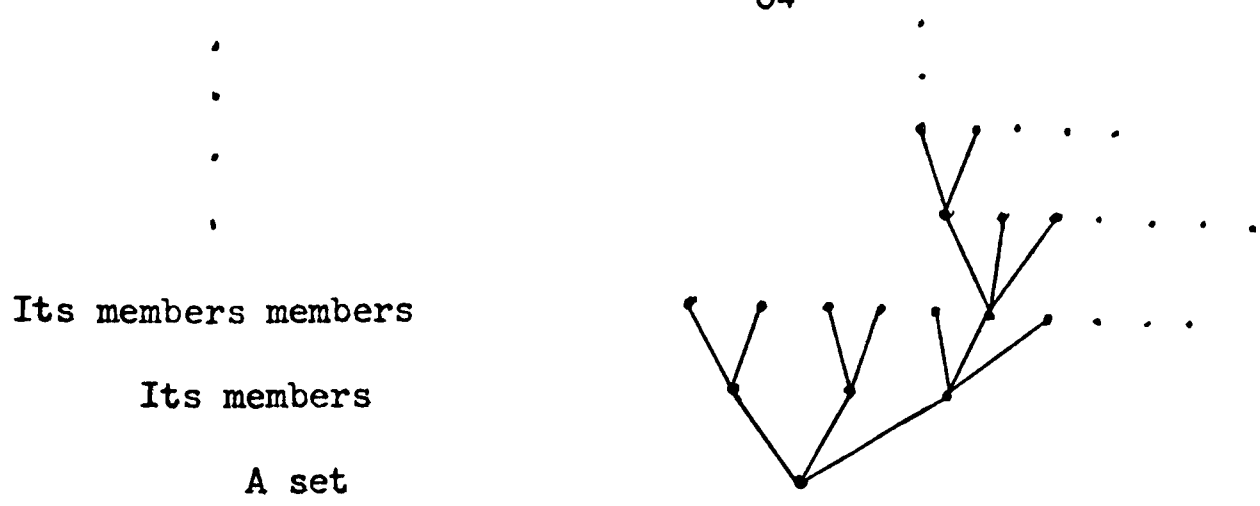
§2. Set theory

We have already used the word set, for objects of M-L, but I shall argue that there is no type of sets in the theory. In 2.4.3 there was the type of 'effectively enumerable sets', but this was just the type of functions N→N with a little ε-structure. In 4.3.1. we shall use the word for a type with an equivalence relation on it; but this however is no real notion of a set as types are disjoint and so have no ∈-structure i.e. ∈ is nothing like 'is a member of a set'. It may be possible to relate sets and types in some other way, but I shall not undertake such a philosophical task here. It is not possible to deal with sets in the way we dealt with reals.

A more useful approach would be to introduce a type of sets with relations ε and ≡ on it. by inductive definition. In this way a set would be defined by those operations which appear in the inductive definition of the type of sets; we could think of these as neo-gödelian operations. Unfortunately attempts I have made in this direction have led to a clumsy syntax and long proofs of, even the most straightforward axiom of set theory. This approach, though, might lead to a more reasonable intuitionistic set theory than merely basing ZF on Intuitionistic rather than Classical Predicate Calculus.

The limited aim of this section, then, will be to model most of intuitionistic set theory in M-L; where the modelling is of course neither classical nor boolean valued. The construction I shall use is based on that of Zucker [1971], in the theory of Ordinals, and of Cole [1971], in the theory of the Topos.

The idea is that a set is a well-founded tree and that being a member is being an immediate proper subtree (which is trivially well-founded and so itself always a set). Pictorially:



More precisely the members are the subtrees up to ≡ isomorphism; which is also identity between sets. The difference between this work and Zucker/Coles will be the constructive rigour of M-L.

4.2.1. Language

We require first a 'type of sets' U, - the domain of the modelling - over which to interpret set variables. But we need some notation: for $s,t \in \tilde{N}$, the type of finite sequences of natural numbers, and $T \in \tilde{N} \to V_o$

$s \subset t =_{df} \forall n \leq lh(s).(s)_n=(t)_n$

$s \subset t = \exists n.\ t=s^* n$

$T_s =_{df} \lambda x.T(s^*x)$

Then the domain, U, is the type of effectively enumerable trees, defined by:

$U =_{df} \Sigma T \in \tilde{N} \to V_o.T(<>)$

$\quad \& \ \Pi s,t \in \tilde{N}.s \subset t \ \& \ T(t) \to T(s).$

$\quad \& \ \Pi f \in N \to N.\Sigma n \in N.\ \sim T(\bar{f}(n))$

Atomic formulae of identity and membership are interpreted using the definitions (for $T,S \in U$);

$T \equiv S =_{df} \Sigma f \in (\Sigma x \in \tilde{N}T(x) \to \Sigma y \in \tilde{N}S(y)).$

$\Sigma g \in (\Sigma y \in \tilde{N}S(y) \to \Sigma x \in \tilde{N}T(x)).$

$f(<>) = g(<>) = <>$

$\quad \& \ \Pi st \in (\Sigma x \in \tilde{N}S(x)).s \subset t \to g(s) \subset g(t).$

$\quad \& \ \Pi st \in (\Sigma y \in \tilde{N}T(y)).s \subset t \to f(s) \subset f(t).$

$S \varepsilon T =_{df} \Sigma n \in N.T(<n>)\ \&\ S \equiv T_{<n>}$.

Note that $\equiv$ and $\varepsilon \in U \times U \to V_o$.

4.2.2 Axioms

4.2.2.1. Those concerning equality

The proof that $\equiv$ is an equivalence relation is easily obtained, as are proofs of

$$T \equiv S \to \Pi v \in U.T \varepsilon v \leftrightarrow S \varepsilon v.$$

$$T \equiv S \to \Pi w \in U.w \varepsilon T \leftrightarrow w \varepsilon S.$$

As for the axiom

$$\Pi v \in U.T \varepsilon v \leftrightarrow S \varepsilon v. \to T \equiv S$$

take the set $\{T\}$ defined by:

$$\{T\}(<>) =_{df} T$$

$$\{T\}(n*\sigma) =_{df} T(\sigma)$$

Then clearly $\{T\} \in U$ and $S \varepsilon \{T\} \Rightarrow S \equiv T$

N.B. in future when defining a set A, I will assume that $A(<>) =_{df} T$ and consider $A(\sigma)$ only where $\sigma \neq <>$. To this end I need the following notation

$$-\sigma =_{df} <\sigma_1, \sigma_2, \ldots, \sigma_{lh(\sigma)}>$$

$$=\sigma =_{df} <\sigma_2, \sigma_3, \ldots, \sigma_{lh(\sigma)}>$$

The non-trivial identity axiom to check is extentionality:

$$\Pi v \in U.\ v \varepsilon T \leftrightarrow v \varepsilon S. \to T \equiv S$$

Assuming the antecedant, in particular:

$$\forall n.T(<n>)\ \&\ \exists m.S(<m>)\ \&\ T_{<n>} \equiv S_{<m>}.$$

$$\forall m.S(<m>)\ \&\ \exists n.T(<n>)\ \&\ T_{<n>} \equiv S_{<m>}.$$

So given a $\sigma \in \tilde{N}$ such that $T(\sigma)$ we have $T(\langle\sigma_o\rangle)$ and hence an $\langle m\rangle \in \tilde{N}$ such that $S(\langle m\rangle)$ and $T_{\langle\sigma_1\rangle} \equiv S_{\langle m\rangle}$.

Then we can define $F \in \Sigma x \in \tilde{N}.T(x) \to \Sigma y \in \tilde{N}.S(y)$ by:

$$F(\sigma) =_{df} \langle m\rangle * f(-\sigma) \quad \text{where} \quad f \text{ is from } T_{\langle\sigma_o\rangle} \equiv S_{\langle m\rangle} \;\&$$

$$f \in \Sigma x \in \tilde{N}.T_{\langle\sigma_o\rangle}(x) \to \Sigma y \in \tilde{N}.S_{\langle m\rangle}(y)$$

and define $G \in \Sigma y \in \tilde{N}.S(y) \to \Sigma x \in \tilde{N}.T(x)$ similarly, to get $T \equiv S$.

4.2.2.2. Those concerning the finite set structure.

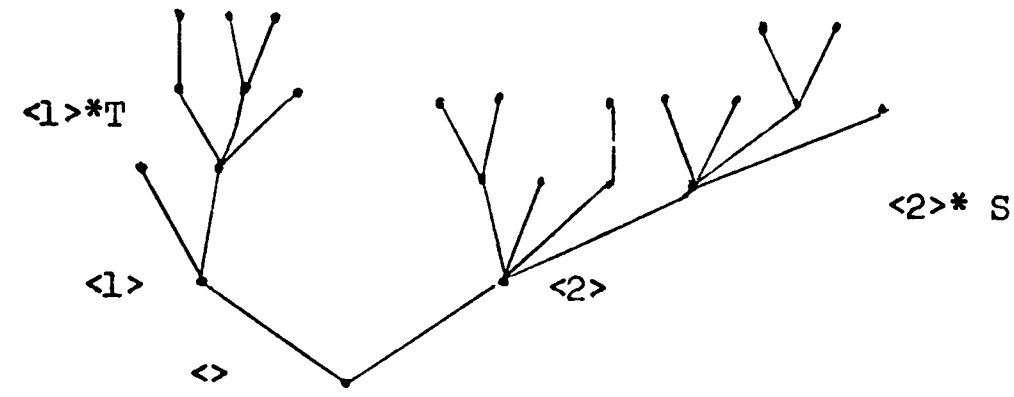
a) the empty set

$$\emptyset(\sigma) =_{df} \sigma = \langle\rangle$$

b) the pair set

$$\{S,T\}(\sigma) =_{df} (\sigma_o = 1 \;\&\; S(-\sigma)) + (\sigma_o = 2 \;\&\; T(-\sigma))$$

pictorially:



c) the union set

$$\cup S(\sigma) = S(\langle \sigma_{oo}\ \sigma_{o1}\rangle * -\sigma)$$

4.2.2.3. Δ_o-comprehension

We check the finite version of this axiom

a) cartesian product

$$S \times T(\sigma) =_{df} (lh(\sigma)=1 \;\&\; T(\langle\sigma_{o,o}\rangle) \;\&\; S(\langle\sigma_{o,1}\rangle))$$

$$+ \quad (lh(\sigma)>1 \;\&\; \{T_{\langle\sigma_{o,o}\rangle}, \{T_{\langle\sigma_{o,o}\rangle}, S_{\langle\sigma_{o,1}\rangle}\}\}(-\sigma))$$

b) the conditional set , $R \Rightarrow S$

i.e. $\{x \varepsilon \mathfrak{T} / \; x \varepsilon R \rightarrow x \varepsilon S\}$

$(R \Rightarrow S)(\sigma) =_{df} T(\sigma) \;\&\; (T_{<\sigma_i>} \equiv R \rightarrow T_{<\sigma_i>} \equiv S)$

c) the intersection of a set

$\bigcap T(\sigma) =_{df} \Sigma n,m \in N.T(<nm>*\sigma$

$\& \; \Pi n_1 \in N.\Sigma \; m_1 \in N.T(<n_1 m_1>) \quad \& \quad T_{<nm>} \equiv T_{<n_1 m_1>}$

Before proceeding to the other clauses I need the following:

T is a singleton $=_{df} \exists n.T(<n>)$ &

$\forall n,m.T(<n>) \;\&\; T(<m>) \rightarrow T_{<n>} \equiv T_{<m>}$.

T is a doubleton $=_{df} \exists n,m.T(<n>) \;\&\; T(<m>) \;\&\; T_{<n>} \neq T_{<m>}$

$\& \; \forall k.T(<k>) \rightarrow T_{<k>} \equiv T_{<n>} + T_{<k>} \equiv T_{<m>}$.

d) the ε1T set

$(\varepsilon 1T)(\sigma) =_{df} T \times T(\sigma) \;\&\; \cup T_{<\sigma_o>}$ is a singleton & $UUT_{<\sigma_o>} \varepsilon \; UUT_{<\sigma_o>}$

$+ \; T_{<\sigma_o \sigma_1>}$ is a singleton & $\exists n.T(<\sigma_o n>)$

& $T_{<\sigma_o n>}$ is a doubleton & $UT_{<\sigma_o \sigma_1>} \; \varepsilon \; U(T_{<\sigma_o n>} - T_{<\sigma_o \sigma_1>})$.

$+ \; T_{<\sigma_o \sigma_1>}$ is a doubleton & $\exists m.T(<\sigma_o m>)$

& $T_{<\sigma_o m>}$ is a singleton & $UT_{<\sigma_o m>} \; \varepsilon \; U(T_{<\sigma_o \sigma_1>} - T_{<\sigma_o n>})$.

$+ \; lh(\sigma)=1 \;\&\; \exists k.T(\sigma*<k>)$ & ditto holds of $\sigma*<k>$.

Again, before proceeding define the set of ordered pairs in a set:

$T^*(\sigma) =_{df} T(\sigma) \;\&\; (\cup T_{<\sigma_o>}$ is a singleton + ($T_{<\sigma_o>}$ is a doubleton &

$(T_{<\sigma_o \sigma_1>}$ is a singleton & $\exists n.T(<\sigma_o n>)$ & $(T_{<\sigma_o n>} - T_{<\sigma_o \sigma_1>})$ is singleton

& $T_{<\sigma_o n>}$ is doubleton

$+ \; T_{<\sigma_o \sigma_1>}$ is a doubleton & $\exists m.T(<\sigma_o m>)$ & $T_{<\sigma_o m>}$ is singleton &

$(T_{<\sigma_o \sigma_1>} - T_{<\sigma_o m>})$ is singleton))

e) the second projection set

$$E(T)(\sigma) =_{df} T^*(\langle\sigma_{o,o}\ \sigma_{o,1}\ \sigma_{o,2}\rangle^*\text{-}\sigma)\ \&\ (\cup T^*_{\langle\sigma_{o,o}\rangle}\ \text{is a singleton}$$

$$+\ \exists m.T^*(\langle\sigma_{o,o}m\rangle)\ \&\ T^*_{\langle\sigma_{o,o}m\rangle}\ \text{is singleton}\ \&$$

$$T^*_{\langle\sigma_{o,o},\sigma_{o,1}\rangle}\ \text{is doubleton}\ \&$$

$$T^*_{\langle\sigma_{o,o},\sigma_{o,1},\sigma_{o,2}\rangle}\ \varepsilon\ T^*_{\langle\sigma_{o,o},\sigma_{o,1}\rangle}\ \text{-}T^*_{\langle\sigma_{o,o}m\rangle})$$

f) the permutation of variables should now be straightforward. I demonstrate the first case - commutability.

$$c(T)(\sigma) =_{df} T^*(\sigma)\ \&\ (T^*_{\langle\sigma_o\sigma_1\rangle}\text{is a doubleton or}\ T^*_{\langle\sigma_o\rangle}\ \text{is singleton}$$

$$\text{or}\ T^*_{\langle\sigma_o\sigma_1\rangle}\ \text{is singleton}\ \&\ \exists n.T^*(\langle\sigma_o n\rangle).T^*_{\langle\sigma_o n\rangle}\ \text{is doubleton}$$

$$\&\ T^*_{\langle\sigma_o n\sigma_2\rangle}\ \varepsilon\ T^*_{\langle\sigma_o n\rangle}\ \text{-}T^*_{\langle\sigma_o\sigma_1\rangle}\ \&\ T^*_{\langle\sigma_o n\sigma_2\rangle}\ (\equiv\sigma)$$

4.2.2.4. the axiom of infinity

First define the natural numbers:

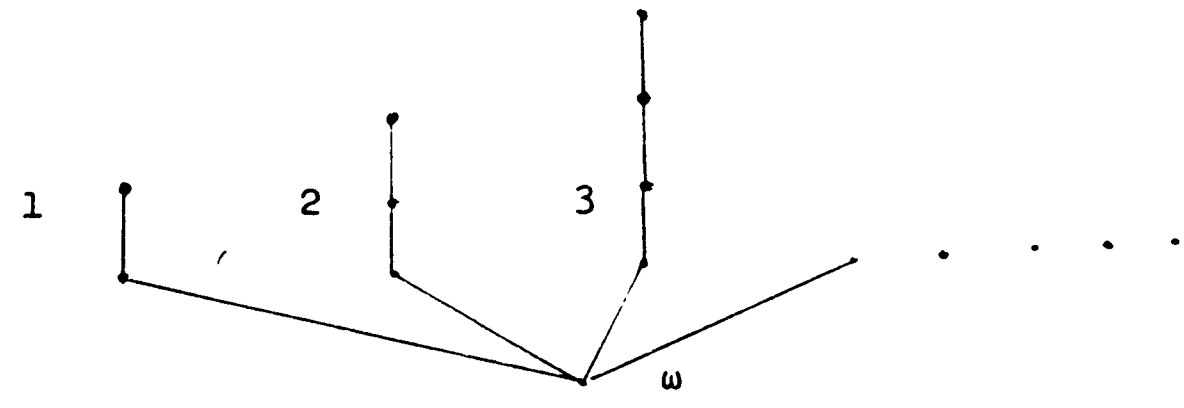
$$n(\sigma) =_{df} lh(\sigma)=n\ \&\ \forall m\leq n,\sigma_m=n$$

and note that $n \varepsilon n+1$

and the infinite set ω is defined by:

$$\omega(\sigma) =_{df} lh(\sigma) =\sigma_o+1\ \&\ \forall m\leq lh(\sigma).\sigma_m=\sigma_o$$

pictorially:



And it is clear that:

$$1\varepsilon\omega \;\&\; \forall x\in U.x\varepsilon\omega \rightarrow \{x\}\varepsilon\omega.$$

and moreover that ω is the smallest such set. Hence induction over ω is possible.

4.2.2.5. The axiom of foundation; in the form

$$\sim\Sigma f\in N\rightarrow U.\Pi n\in N.f(n)\varepsilon T \;\&\; f(n+1)\varepsilon f(n).$$

Assume the antecedant, that there is such an f and use the notation:

$m_o =_{df} \mu m.f(0)\equiv T_{<m>}$

$m_i =_{df} \mu t.f(i)\equiv f(i\dot{-}1)_{<t>}$

$f_o =_{df}$ the unique $f \in \Sigma x\in\tilde{N}.f(0)(x) \rightarrow \Sigma y\in\tilde{N}.T_{<m>}(y)$

such that isomorphism

$f_i =_{df}$ the unique $f \in \Sigma x\in\tilde{N}.f(i)(x) \rightarrow \Sigma y\in\tilde{N}.f(i-1)(y)$

such that isomorphism

Then define $g \in N \rightarrow N$ by:

$$g(n) =_{df} \langle f_o(\langle m_o\rangle * (\ldots f_{n-1}(\langle m_{n-1}\rangle * f_n(\langle m_n\rangle))\ldots)_{lh(f_o(\langle m_o\rangle * (\ldots)))}\rangle$$

and note that $\Pi m\in N.T(\bar{g}(m))$ which contradicts the well-founded T.

In addition to set theory we can model

4.2.2.6. The axiom of choice in the form

$$\Sigma a\in U.a\varepsilon T \;\&\; \Pi t\in U.t\varepsilon T \;\&\; \Sigma y_t\in U.y_t\varepsilon t.$$

$$\&\; \Pi s,t\in U.s,t\varepsilon T \rightarrow s\cap t\equiv\emptyset$$

$$\rightarrow \Sigma f\in T\Rightarrow UT.\Pi x\in U.x\varepsilon T \;\&\; f(x)\varepsilon x$$

where $T\Rightarrow UT$ is the type of maps from $\Sigma x\in U.x\varepsilon T$ to $\Sigma y\in U.y\varepsilon UT$ with respect to the relation $\equiv$.

In this case I need only find a function from $\Sigma x\in U.x\varepsilon T$ to $\Sigma y\in U.y\varepsilon UT$ as all members of T are distinct.

Applying the axiom of choice of M-L to the antecedant gives us a map $f \in T \Rightarrow UT$ by:

$$f(t) =_{df} y_t \varepsilon t$$

4.2.2.7. The axiom of transitive closure, that every set has one. Define it by coding the position of the node in the original tree as the initial node of the new tree:

$$cl\,T(\sigma) =_{df} T(\langle \sigma_{o,o}, \sigma_{o,1}, \ldots, \sigma_{o,lh(\sigma_o)} \rangle * - \sigma)$$

§3. Category theory

A search through any standard text of category theory - for example, MacLane [1971] - will not produce an example of a non-constructive proof. It will not be surprising, therefore, that category theory can be completely formalized inside M-L; indeed it was the possibility of this that motivated Martin-Löf to develop the theory of types. In what follows I shall make some of the more simple definitions and show how to use them to make Kripke models of M-L.

4.3.1. Some definitions

A category of order (n,m) is a 5-tple comprising: a type of universe V_n - the domain of objects; a function taking a pair of objects in the domain to a type of universe V_m - the type of homomorphisms between these objects; a function taking an object to its identity homomorphism; a function taking three objects to the composition operation on their appropriate homomorphism types; and a function taking a pair of objects to an equivalence relation on their homomorphism type. And in addition a proof that these types and functions have the intended properties.

Formally, the categories of order (n,m) form a type, cat defined by

$$\mathrm{cat}_{(n,m)} =_{df} \Sigma X \epsilon V_n . \Sigma H \epsilon X \times X \to V_m . \Sigma U \epsilon (\Pi Y \epsilon X . H(Y,Y)).$$

$$\Sigma V \epsilon (\Pi Y,Z,T \epsilon X . H(Y,Z) \times H(Z,T) \to H(Y,T)).$$

$$\Sigma W \epsilon (\Pi Y,Z \epsilon X . H(Y,Z) \times H(Y,Z) \to V_m).$$

$$A[H,W] \;\&\; B[H,W,V] \;\&\; C[H,W,V,U]$$

where A, B and C are defined as follows: writing W(A,B,f,g) as $f \equiv_{AB} g$, V(A,B,f,g) as $f \circ g$ and U(A) as 1_A:

$$A[H,W] =_{df} \Pi A,B \epsilon X . \Pi f,g,h \epsilon H(A,B).\ f \equiv_{AB} f \;\&\; (f \equiv_{AB} g \to g \equiv_{AB} f) \;\&$$

$$(f \equiv_{AB} g \;\&\; g \equiv_{AB} h) \to f \equiv_{AB} h.$$

$$B[H,W,V] =_{df} \Pi A,B,C,D \epsilon X . \Pi f \epsilon H(A,B) . \Pi g \epsilon H(B,C) . \Pi h \epsilon H(C,D) . f \circ (g \circ h) \equiv_{AD} (f \circ g) \circ h.$$

$$\;\&\; \Pi f,g \epsilon H(A,B) . \Pi h \epsilon H(B,C).\ f \circ h \equiv_{AC} g \circ h \to f \equiv_{AB} g$$

$$\;\&\; \Pi h \epsilon H(A,B) . \Pi f,g \epsilon H(B,C).\ h \circ f \equiv_{AC} h \circ g \to f \equiv_{BC} g$$

$$C[H,W,V,U] =_{df} \Pi A,B \epsilon X . \Pi f \epsilon H(A,B).\ f \circ 1_B \equiv_{AB} 1_A \circ f \equiv_{AB} f.$$

For the model forming purposes mentioned above we will only need to have n=m and so write $\mathrm{cat}_{(n,n)}$ as cat_n; and it follows that $\mathrm{cat}_n \in V_{nH}$. Further conventions are that for $C \in \mathrm{cat}_n$ we refer to p(C) as ob(C), and p(q(C)) as hom_C.

A set of order is a category of order n whose homomorphism types are empty, or have an identity . Formally the sets of order n are the type defined by:

$$\Pi x \in ob(X) . \Pi f,g \in hom_X(x,x).$$

$$set_n =_{df} \Sigma X \epsilon cat_n . \Pi Y,Z \epsilon ob(X).\ Y \neq Z \to (hom_X(Y,Z) \to 1), \qquad f \equiv_{xx} g.$$

And from set_n we can form a category of order n+1, a category of sets, set_n^*. This is defined by:

$$ob(set_n^*) =_{df} set_n$$

$$hom_{set_n^*}(A,B) =_{df} ob(p(A)) \to ob(p(B))$$

$$f \equiv_{AB} g =_{df} f =_{ext} g$$

$$f_{\circ}g =_{df} \lambda x.g(f(x))$$

$$1_A =_{df} \lambda x.x$$

(In what follows we write ob(A) for ob(p(A)) and hom_A for $hom_{p(A)}$ where $A \in set_n$).

A functor from category A to category B is a pair comprising: a function from ob(A) to ob(B) and a function taking the homomorphism type of a pair of objects in ob(A) to the homomorphism type of their images under the function from ob(A) to ob(B). Together with a proof that they satisfy certain conditions. Formally the functors between $A \in cat_n$ and $B \in cat_n$ form a type fun(A,B) or B^A which is defined by:

$$B^A =_{df} \Sigma F \in ob(A) \to ob(B).\Sigma G \in (\Pi X,Y \in ob(A).hom_A(X,Y) \to hom_B(F(X),F(Y))).$$

$$\Pi X,Y,Z \in ob(A).\Pi f \in hom_A(X,Y).\Pi g \in hom_A(Y,Z).$$

$$G(X,Z,f_{\circ}g) \equiv_{F(X)F(Z)} G(X,Y,f)_{\circ}G(Y,Z,g).$$

$$\& \Pi X \in ob(A).1_{F(X)} \equiv_{F(X)F(X)} G(X,X,1_X)$$

If $F \in B^A$ then we write F_X for p(F)(X) and F_{XY} for p(q(F))(X,Y).

Similarly, the natural transformations between two functors $F,G \in B^A$ form a type, nat(F,G), defined by:

$$nat(F,G) =_{df} \Sigma R \in (\Pi X \in ob(A).hom_B(F_X,G_X)).$$

$$\Pi X,Y \in ob(A).\Pi f \in hom_A(X,Y).F_{XY}(f)_{\circ}R(Y) \equiv_{F_XG_Y} R(X)_{\circ}G_{XY}(f).$$

and from B^A we can form a category of order max(n,m), the functor category, fun*(A,B), which is defined as follows:

$$ob(fun^*(A,B)) =_{df} fun(A,B)$$

$$hom_{fun^*(A,B)}(F,G) =_{df} nat(F,G)$$

$$R \equiv_{FG} S =_{df} \Pi X \in ob(A).R(X) \equiv_{F_XG_X} S(X)$$

$$R_{\circ}G =_{df} (\lambda X.R(X)_{\circ}S(X), \text{ obvious proof})$$

$$1_F = (\lambda X.1_{F_X}, \text{ obvious proof})$$

A special set, the one point set is distinguished by *; and can be defined by taking $ob(p(*)) =_{df} T$ or $T\times(V_{n-1}=V_{n-1})$ according as * is the one point set of order 0 or n. A special functor, the constant functor, is distinguished by ΔC; and defined by taking $(\Delta C)_X =_{df} C$, for each X.

All the universal objects can be constructed, and for the model theoretic applications to follow, we shall show how to construct the product and coproduct of a family of sets. Given $J \in set_n$, an object, F, of type set_m^{*J} is a family of sets of order m, indexed by J. The product of that family over J, $\prod_J F$, is defined as follows:

$$ob(\prod_J F) =_{df} \Pi X \in ob(J).F_X$$

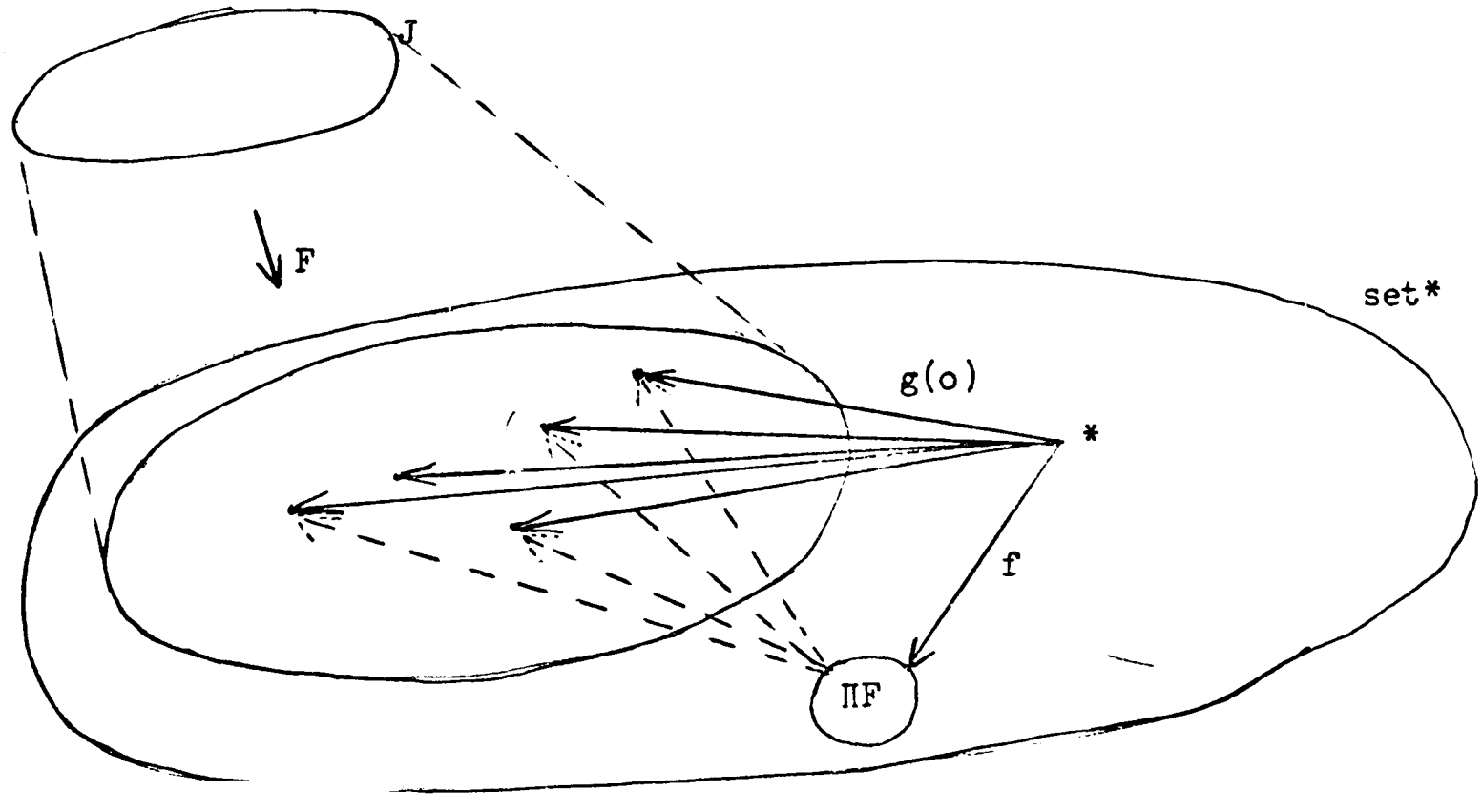
$$hom_{\prod_J F}(f,g) =_{df} f=g$$

$$x \equiv_{fg} y =_{df} x=r(f)$$

$$1_f = r(f)$$

and by definition of $hom_{\prod_J F}$, $\prod_J F$ is a set; $(\prod_J F, \lambda x.x) \in set_{max(n,m)}$.

For every product, ΠF, there are projections $p_X \in \prod_J F \to F_X$ given by $p_X(f) =_{df} f(X)$. And for every cone of functions, $g \in nat(\Delta^*,F)$ there is a function $f \in ob(*) \to ob(\prod_J F)$ which makes the following diagram commute:



Similarly we can form the coproduct of a family of sets over an index set k:

$$ob(\coprod_J F) =_{df} \Sigma X \in ob(J).F_X$$

$$hom_{\coprod_J F}(u,v) =_{df} u=v$$

$$x \equiv_{uv} y =_{df} x=r(u)$$

$$x \circ y =_{df} \text{proof of transitivity}$$

$$1_u =_{df} r(u)$$

4.3.2. Kripke models and M-L

The motivation for considering Kripke models in a study of Martin-Löf type theory is two-fold. Firstly, if we could set up the machinery for constructing Kripke models of HA and IPC in M-L, then it might be possible to give constructive proofs of some of the usual model theoretic results for these models. For example, we might be able to give a constructive proof of completeness for Kripke models. Secondly, a Kripke-style model of M-L itself might be useful for consistency applications, or interesting in its own right.

With regard to a general machinery for Kripke models of HA and IPC, it should be noted that there are no atomic formulae in M-L. The function constant way of formalizing the type theory makes it clear that $a=_C b$ is not atomic. The interpretation of $a=_C b$ must be a function of the interpretation of C. This difficulty could be overcome by adding to the language of M-L enough atomic formulae.

The following development of Kripke models for M-L itself also provides the machinery for Kripke models of HA and IPC as a special case.

4.3.2.1. Kripke models of M-L

We begin with a cat of order 0 which is partially ordered; that is a set of type po where:

$$po =_{df} \Sigma X \in set_o.\Pi x,y \in ob(X).\Pi f,g \in hom_x(x,y).f \equiv_{xy} g.$$

$$\& \Sigma z \in ob(X).\Pi x \in ob(X).hom_x(z,x).$$

(For $y \in po$, write $hom_{p(y)}$ as $\leq$ and $p(q(q(Y)))$ as 0)

Then we have all the underlying structure that is needed. Given $I \in po$ we assign for each typesymbol $A \in V_n$ a functor $\bar{A} \in set_n^I$. And writing $i \models A$ for $ob(\bar{A}_i)$ we have the usual:

$$\Pi i,j \in ob(I).\ i \models A \ \& \ i \leq j \rightarrow j \models A$$

The assignment provides a cumulative hierarchy (in some sense) of types for each typesymbol, which are to be the parameters of the appropriate type. As will become clear the assignment for typesymbols such as N and V_n is forced upon us and the assignment for complex typesymbols is determined by the fact that the typesymbols are also the propositions. As suggested by the notation '$i \models A$' the assignment to a typesymbol is also part of the definition of the usual turnstyle relation between members of the partially ordered cat and the propositions.

Bearing these considerations in mind, it will become clear that Kripke models for M-L will be a weak tool for consistency results. As Smorynski [1973] puts it, we shall have to rely on the 'geometry' of the underlying set.

Given a partially ordered cat, $I \in po$, we form the Kripke model, $K(I)$, over I as follows:

i) $Typ_n =_{df} fun(\perp, set_n^*)$

ii) for each $\Phi \in Typ$:

$$Obj_n(\Phi) =_{df} ob(\Phi_o)$$

iii) the interpretation of the terms is given over the definition of $\in$ and conv as usual. For each closed term $a \in A$ assign an $|a| \in Obj(|A|)$; but open terms need more. We shall have to strengthen the induction hypothesis.

Suppose that for $x_1 \in A_1$, $x_2 \in A_1[x_1],\ldots,x_n \in A_n[x_1,\ldots,x_{n-1}]$ there is a term $b[x_1,\ldots,x_n] \in B[x_1,\ldots,x_n]$ and also the typesymbols $A_1 \in V_{k_1}$, $A_2[x] \in V_{k_2},\ldots, A_n[x_1,\ldots,x_{n-1}] \in V_{k_n}$, $B[x_1,\ldots,x_n] \in V_k$. Then assign the following functions, whose ranges are functors:

$$\bar{A}_2 \in \Pi x \in ob(I).fun(|A_1|_x, fun^*(I, set^*_{k_2}))$$

$$\bar{A}_3 \in \Pi x \in ob(I).\Pi y \in ob(|A_1|_x).fun(\bar{A}_2(x)_y, fun^*(I, set^*_{k_3}))$$

$$\vdots$$

$$\bar{B} \in \Pi x \in ob(I).\Pi y_1 \in ob(|A_1|_x)\ldots \Pi y_{n-1} \in ob(\bar{A}_{n-1}(x,y_1,\ldots,y_{n-2})).$$
$$fun(\bar{A}_n(x,y_1,\ldots,y_{n-1}), fun^*(I, set^*_k))$$

in such a way that for $\xi_1 \in Obj(|A_1|)$:

$$\bar{A}_2(0)_{\xi_1} = |A_2[\underline{\xi_1}]|$$

for $\xi_1 \in Obj(|A_1|)$ and $\xi_2 \in Obj(|A_2[\underline{\xi_1}]|)$:

$$\bar{A}_3(0)(\xi_1)_{\xi_2} = |A_3[\underline{\xi_1},\underline{\xi_2}]|$$

etc. And assign the function:

$$\bar{b} \in \Pi x \in ob(I).\Pi y \in ob(|A_1|_x)\ldots\Pi z \in ob(\bar{A}_n(x,y,\ldots)).\bar{B}(x,y,\ldots)$$

in such a way that for $\xi_1 \in Obj(|A_1|),\ldots,\xi_n \in Obj(|A_n[\underline{\xi_1},\ldots,\underline{\xi_{n-1}}]|)$:

$$\bar{b}(0,\xi_1,\ldots,\xi_n) = |b[\underline{\xi_1},\ldots,\underline{\xi_n}]|$$

We now give the definition of $|.|$ groupwise and in each case omit that of $\bar{.}$ which is just obtained from the induction hypothesis, in the obvious way.

Group A

$$|V_n| =_{df} \Delta fun^{*}(I,set_n^*)$$

and $fun^*(I,set_n^*) \in set_{n+1}$ so $|V_n| \in fun(I,set_{n+1}^*) =_{df} Obj(|V_{n+1}|)$

Group B

i) $|\Pi x \in A.B[x]|_i =_{df} \Pi j \in ob(I).\ i \leq j \to ob(\underset{A_j}{\Pi} \bar{B}(j))$.

$$|\Pi x \in A,B[x]|_{ij} =_{df} \lambda f.f1_j$$

ii) $$|\lambda x.b[x]| =_{df} \bar{b}(0)$$

iii) $$|f(a)| =_{df} |f|(|a|)$$

Group C

i) firstly define an object $M \in set$, as follows:

$$ob(M) =_{df} N$$

$$hom_{\mu}(n,m) =_{df} \begin{cases} T & \text{if } n=m \\ \perp & \text{o.w.} \end{cases}$$

$$x \equiv_{nm} y =_{df} x=1^*$$

$$1_n =_{df} 1^*$$

and then:

$$|N| =_{df} \Delta M$$

ii) $|0| =_{df} 0$ and $|(s;a)| =_{df} s(|a|)$

iii) $|rec_{a,b}[c]| =_{df} rec_{|a|,|b|}[|c|]$

Group D

i) $|a =_c b|_i =_{df} \bar{C}_{oi}(|a|) =_{\bar{c}_i} \bar{C}_{ci}(|b|)$

$|a =_c b|_{ij} =_{df} \lambda x.r(\bar{C}_{oj}(|a|))$

ii) $|r(a)| =_{df} r(|a|)$

iii) $|id_b[c]| =_{df} id_{|b|}[c]$

4.3.2.2 Kripke models of HA and IPC

The definition of ' $\models$ ' in 4.3.2.1. is a generalization of the usual definition of the relation between I and the propositions of the language of HA or IPC. We could use this definition to make Kripke models of HA and IPC, provided the language of M-L was extended to include atomic formulae, as explained above. Otherwise we could proceed as follows: given a partially ordered set I, an assignment to the atomic formulae of certain functors in set_o^{*I} and a distinguished functor $D \in set_o^{*I}$: interpret the complex formulae of HA or IPC in the language of M-L in the obvious way. The resulting model K(I) is a model of HA or IPC under this interpretation if we take the free variables of the interpretations of the open formulae as ranging over D. That is if we take D_i as the parameters for each $i \in ob(I)$.

Chapter 5

THE ORDINAL OF M-L

Much work has been done towards determining the proof theoretic ordinal of M-L. Peter Hancock [1973] has shown that the ordinal of M-L with all the universe typesymbols is at least Γ_0; and in fact conjectures there that it is exactly Γ_0. Subsequent work by Aczel [1974] tends to confirm the conjecture.

Whilst the presence of the universe typesymbols pushes up the size of the ordinal, still the theory without universes has ordinal $\phi_{\varepsilon_0}(0)$. The main work of this chapter will be to show that the theory gets its strength mainly from the rules for cartesian products, that is from the presence of functions. This will be done by showing that the theory without both cartesian products and the universe typesymbols has a small proof theoretic ordinal.

I call the residue finitist type theory (ftt). And although it does not correspond to any particular philosophical position, it can be thought of as a formalization of finitism. The reason is that the only functions that we can have signs for have finite domains and are given pointwise. Also, ftt, as will become clear, satisfies Tait's suggestion [1968] that the only finitist numerical open terms are primitive recursive.

§1. Finitist Type Theory

We cannot formalize finitist type theory by simply dropping the rules of group B and those containing V_i from M-L. Firstly, in M-L we relied on the group B rules to define the finite types N_k and the disjoint union of two types A+B; see 1.5.1. In order to have these in ftt, we shall have to add them to the system primitively. In fact we will not need to add all of them primitively; some will be definable. Secondly, in M-L the generation of terms and typesymbols was simultaneous, because a typesymbol was also a term. Without the V_i, we will have to separate the two processes.

5.1.1. Language, of ftt

We begin with the variables and function constants of M-L; less the distinguished 0-ary function constants V_i for $i>0$ and the auxiliary signs Π, λ and ap. And in addition the distinguished 0-ary function constant N_1 and the auxiliary signs: +, i, j, dis, 1*, r_o, r_1.

Associations are made as before, but with the additions that: N_1 is associated with V_o, 1* is associated with N_1 and + is associated like Σ, i and j like r, dis like rec and r_o,r_1 like λ. A string and a formal expression are defined the same way as in M-L.

5.1.2. The definition of ϵ and conv

The typesymbols are no longer terms, if we are to have no universes. So we must give separate rules of typesymbol and term formation. These are respectively group H and groups A to G' below.

i) N and N_1 are typesymbols and if f is a function constant introduced in Group H with index typesymbols/value string:

$$A_1,A_2[x_1],\ldots,A_n[x_1,\ldots,x_{n-1}]/V_o$$

and $a_1 \epsilon A_1$, $a_2 \epsilon A_2[a_1],\ldots,a_n \epsilon A_n[a_1,a_2,\ldots,a_{n-1}]$ then:

$(f;a_1,a_2,\ldots,a_n)$ is a typesymbol.

ii) If $a \epsilon A$ for some typesymbol A then a is a term.

Group A As in M-L except that the function constant is introduced in 6,7,10, 11,14,15,17,18,20 or 23 below. And without $V_n \epsilon V_{n+1}$.

Group C 6), 7) and 8 of M-1

Group D 10),11) and 12) of M-L

Group E 14),15) and 16) of M-1

Group F

17) +-introduction. Introduce n+1-ary function constants i and j with index/value typesymbols:

$$A_1,A_2[x_1],\ldots,A_n[x_1,\ldots,x_{n-1}],A[x_1,\ldots,x_n]/A[x_1,\ldots,x_n] + B[x_1,\ldots,x_n]$$

and $$A_1,A_2[x_1],\ldots,A_n[x_1,\ldots,x_{n-1}],B[x_1,\ldots,x_n]/A[x_1,\ldots,x_n] + B[x_1,\ldots,x_n]$$

18) +-elimination. If for $x \in A[x_1,\ldots,x_n]$ and $y \in B[x_1,\ldots,x_n]$, $f[x_1,\ldots,x_n,x] \in C[x_1,\ldots,x_n,(i;x)]$ and $g[x_1,\ldots,x_n,y]$ $C[x_1,\ldots,x_n,(j;y)]$ then introduce the n+1-ary function constant:

$$dis(f[x_1,\ldots,x_n,x],g[x_1,\ldots,x_n,y])$$

with index/value typesymbols:

$$A_1,A_2[x_1],\ldots,A_n,\ A[x_1,\ldots,x_n] + B[x_1,\ldots,x_n] \ /\ C[x_1,\ldots,x_n,z]$$

where $z \in A[x_1,\ldots,x_n] + B[x_1,\ldots,x_n]$.

19) +-conversion. If $a_1 \in A_1,\ldots,a_n \in A_n[a_1,\ldots,a_{n-1}]$, $a \in A[a_1,\ldots,a_n]$ and $b \in B[a_1,\ldots,a_n]$ then:

$$(dis(f,g);a_1,\ldots,a_n,(i;a_1,\ldots,a_n,a)) \text{ conv } f[a_1,\ldots,a_n,a]$$

$$(dis(f,g);a_1,\ldots,a_n,(j;a_1,\ldots,a_n,b)) \text{ conv } g[a_1,\ldots,a_n,b]$$

Group G

20) N_1-introduction. $1^* \in N_1$

21) N_1-elimination. If $a[x_1,\ldots,x_n,1^*] \in C[x_1,\ldots,x_n,1^*]$ then introduce the n+1-ary function constant r_1 with index/value typesymbols:

$$A_1,A_2[x_1],\ldots,A_n[x_1,\ldots,x_{n-1}],N_1 \ /\ C[x_1,\ldots,x_n,z] \ ;z \in N_1$$

22) N_1-conversion. If $a_1 \in A_1,\ldots,a_n \in A_n[a_1,\ldots,a_{n-1}]$ and $b \in N_1$, then:

$$(r_1(a[x_1,\ldots,x_n]);a_1,\ldots,a_n,b) \text{ conv } a[a_1,\ldots,a_n,b]$$

Group G'

23) Introduce the n+1-ary function constant:

$$r_o(C[x_1,\ldots,x_n])$$

with index/value typesymbols:

$$A_1,A_2[x_1],\ldots,A_n[x_1,\ldots,x_{n-1}],(=;0,1)/C[x_1,\ldots,x_n]$$

Group H The rules of typesymbol formation.

24) If $A[x_1,\ldots,x_n]$ and $B[x_1,\ldots,x_n,x]$ are typesymbols where $x \in A[x_1,\ldots,x_n]$ then introduce the n-ary function constant

$$\Sigma(A[x_1,\ldots,x_n],B[x_1,\ldots,x_n,x])$$

with index/value strings:

$$A_1,A_2[x_1],\ldots,A_n[x_1,\ldots,x_{n-1}]/V_o$$

25) If $A[x_1,\ldots,x_n]$ is a typesymbol then introduce the n+2-ary function constant =, with index/value typesymbols:

$$A_1,\ldots,A_n[x_1,\ldots,x_{n-1}],A[x_1,\ldots,x_n],A[x_1,\ldots,x_n]/V_o$$

26) If $A[x_1,\ldots,x_n]$ and $B[x_1,\ldots,x_n]$ are typesymbols then introduce the n-ary function constant:

$$A[x_1,\ldots,x_n] + B[x_1,\ldots,x_n]$$

with argument/value strings:

$$A_1,A_2[x_1],\ldots,A_n[x_1,\ldots,x_{n-1}]/V_o$$

27) If for $x \in N$ and $y \in V_o$, $A[x,\ldots,x_n]$ is a typesymbol

$$\& B[x_1,\ldots,x_n,x,y] \in V_o$$

then introduce the n+1-ary function constant:

$$rec(A[x_1,\ldots,x_n],\ B[x_1,\ldots,x_n,x,y])$$

with index/value typesymbols:

$$A_1,\ldots,A_n[x_1,\ldots,x_{n-1}],N/V_o$$

28) If $a_1 \in A_1,\ldots,a_n \in A_n[a_1,\ldots,a_{n-1}]$, $c \in N$ then

$$(rec(A,B);\ a_1,\ldots,a_n,0)\ conv\ A[a_1,\ldots,a_n]$$

$$(rec(A,B);\ a_1,\ldots,a_n,(s;c))\ conv\ B[a_1,\ldots,a_n,c,(rec;a_1,\ldots,a_n,c)]$$

29) If for $x \in A[x_1,\ldots,x_n]$ & $y \in B[x_1,\ldots,x_n,x],C[x_1,\ldots,x_n,x,y]$ is a typesymbol then introduce the n+1-ary function constant $un(C[x_1,\ldots,x_n,x,y])$ with index/value typesymbols:

$$A_1,\ldots,A_n,(\Sigma(A,B);\ x_1,\ldots,x_n)/V_o$$

30) If $a_1 \in A_1, \ldots, a_n \in A_n$, $d \in A[a_1,\ldots,a_n]$, $e \in B[a_1,\ldots,a_n,d]$ then

$(un(C);a_1,\ldots,a_n,d,e)$ conv $C[a_1,\ldots,a_n,d,e]$

31) If $B[x_1,\ldots,x_n,x]$ is a typesymbol, for $x \in A[x_1,\ldots,x_n]$, then introduce the n+3-ary function constant $id(B[x_1,\ldots,x_n,x])$ with index/value typesymbols:

$$A_1,\ldots,A_n,\ A[x_1,\ldots,x_n],\ (=;x_1,\ldots,x_n,y,z)/V_o$$

32) $(id(B);a_1,\ldots,a_n,c,c,(r;c))$ conv $B[a_1,\ldots,a_n,c]$

33) If for $x \in A[x_1,\ldots,x_n]$ and $y \in B[x_1,\ldots,x_n]$, $F[x_1,\ldots,x_n,x]$ and $G[x_1,\ldots,x_n,y]$ are typesymbols then introduce the n+1-ary function constant:

$$dis(F[x_1,\ldots,x_n,x],G[x_1,\ldots,x_n,y])$$

with index/value typesymbols:

$$A_1,A_2[x_1],\ldots,A_n,A[x_1,\ldots,x_n] + B[x_1,\ldots,x_n]/V_o$$

34) If $a_1 \in A_1,\ldots,a_n \in A_n[a_1,\ldots,a_{n-1}]$, $a \in A[a_1,\ldots,a_n]$ and $b \in B[a_1,\ldots,a_n]$ then:

$(dis(F,G);a_1,\ldots,a_n,(i;a_1,\ldots,a_n,a))$ conv $F[a_1,\ldots,a_n,a]$

$(dis(F,G);a_1,\ldots,a_n,(j;a_1,\ldots,a_n,b))$ conv $G[a_1,\ldots,a_n,b]$

35) If $A[x_1,\ldots,x_n,1^*]$ is a typesymbol then introduce the n+1-ary function constant r_1 with index/value typesymbols:

$$A_1,A_2[x_1],\ldots,A_n[x_1,\ldots,x_{n-1}],N_1/V_o$$

36) If $a_1 \in A_1,\ldots,a_n \in A_n[a_1,\ldots,a_{n-1}]$ and $b \in N_1$, then:

$(r_1(A[x_1,\ldots,x_n]);a_1,\ldots,a_n,b)$ conv $A[a_1,\ldots,a_n,b]$

The rule 23) allows us to derive anything from 0=1. In 1.5.1. this was provable, and we were able to take 0=1 as the empty type. It is not provable in ftt without rule 23); the proof of 1.5.1. made essential use of the rules of Group B.

5.1.3. The simplified formalization

Group A The conversion rules, as before.

Group C

N-introduction

$$0 \in N \qquad \frac{x \in N}{s(x) \in N}$$

N-elimination

$$\frac{z \in N \qquad a \in C[o] \qquad \begin{array}{c} x \in N \quad y \in C[x] \\ \vdots \qquad\quad \vdots \\ b[x,y] \in C[s(x)] \end{array}}{rec_{a,b}[z] \in C[z]}$$

N-conversion

$$rec_{a,b}(0) \text{ conv } a;\ rec_{a,b}(s(c)) \text{ conv } b[c,rec(c)]$$

Group D

Σ-introduction

$$\frac{a \in A \qquad b \in B[a]}{(a,b) \in \Sigma x \in A.B[x]}$$

Σ-elimination

$$\frac{z \in \Sigma x \in A.B[x] \qquad \begin{array}{c} x \in A, \quad y \in B[x] \\ \vdots \qquad\quad \vdots \\ c[x,y] \in C[(x,y)] \end{array}}{un_c[z] \in C[z]}$$

Σ-conversion

$$un_c((a,b)) \text{ conv } c[a,b]$$

Group E

=-introduction

$$\frac{a \in A}{r(a) \in a =_A a}$$

=-elimination

$$\frac{z \in A =_A b \qquad \begin{array}{c} t \in A \\ \vdots \\ b[t] \in C[t,t,r(t)] \end{array}}{id_b[z] \in C[a,b,z]}$$

=-conversion

$$id_b(r(a)) \text{ conv } b[a]$$

Group F

+-introduction

$$\frac{a \in A}{i(a) \in A+B} \qquad \frac{b \in B}{j(b) \in A+B}$$

+-elimination

$$\frac{z \in A+B \qquad \begin{array}{c} x \in A \\ \vdots \\ f[x] \in C[i(x)] \end{array} \qquad \begin{array}{c} y \in B \\ \vdots \\ g[y] \in C[j(y)] \end{array}}{dis_{f,g}[z] \in C[z]}$$

+-conversion

$$dis_{f,g}(i(c)) \text{ conv } f[c]$$

$$dis_{f,g}(j(d)) \text{ conv } g[d]$$

Group G

N_1-introduction

$$1^* \in N_1$$

N_1-elimination

$$\frac{z \in N_1 \qquad a \in C[1^*]}{r_{1,a}(z) \in C[z]}$$

N_1-conversion

$$r_{1,a}(1^*) \text{ conv } a$$

Group G'

$$\frac{z \in 0=1}{r_0(z) \in C}$$

Group H

i) N, N_1 are typesymbols.

ii)
$$\frac{A \text{ is typesymbol} \qquad \begin{array}{c} x \in A \\ \vdots \\ B[x] \text{ is typesymbol} \end{array}}{\Sigma x \in A.B[x] \text{ is typesymbol}}$$

iii)
$$\frac{A \text{ is typesymbol} \qquad a,b \in A}{a =_A b \text{ is a typesymbol}}$$

iv)
$$\frac{A,B \text{ are typesymbols}}{A+B \text{ is a typesymbol}}$$

v)
$$\frac{z \in N \qquad A \text{ is typesymbol} \qquad \begin{array}{c} x \in N \\ \vdots \\ B[x,y] \in V_o \end{array} \qquad \begin{array}{c} y \in V_o \\ \vdots \end{array}}{rec_{A,B}[z] \text{ is a typesymbol}}$$

and $rec_{A,B}[o]$ conv A; $rec_{A,B}[s(c)]$ conv $B[c,rec_{A,B}[c]]$

vi)
$$\frac{z \in \Sigma x \in A.B[x] \qquad \begin{array}{c} x \in A \quad y \in B[x] \\ \vdots \qquad \vdots \\ C[x,y] \text{ is a typesymbol} \end{array}}{un_c[z] \text{ is a typesymbol}}$$

and $un_c((a,b))$ conv $C[a,b]$

vii)
$$\frac{z \in a =_A b \qquad \begin{array}{c} t \in A \\ \vdots \\ B[t] \text{ is a typesymbol} \end{array}}{id_B[z] \text{ is a typesymbol}}$$

viii)
$$\frac{\begin{array}{ccc} z \in A+B & \begin{array}{c} x \in A \\ \vdots \\ F[x] \text{ is a typesymbol} \end{array} & \begin{array}{c} y \in B \\ \vdots \\ G[y] \text{ is a typesymbol} \end{array} \end{array}}{\text{dis}_{F,G}[z] \text{ is a typesymbol}}$$

and $\text{dis}_{F,G}(i(a))$ conv $F[a]$; $\text{dis}_{F,G}(j(b))$ conv $G[b]$

ix)
$$\frac{z \in N_1 \qquad A \text{ is a typesymbol}}{r_{1,A}(z) \text{ is a typesymbol}}$$

and $r_{1,A}(1^*)$ conv A.

5.1.4. Definition of the finite types

We can now define the finite typesymbols N_k for $k \geq 2$ and they clearly satisfy the rules of group G_k:

$$N_k =_{df} T + N_{k-1}$$

and for example: $1_2 =_{df} j(1_2)$, $2_3 =_{df} j(2_2)$, $3_3 = i(1^*)$

so $1_3 = j(i(1^*))$, $2_3 = j(j(i(1^*)))$ and the definition of r_3 is obvious.

5.1.5. Some theorems of ftt

Whilst we do not have Leibniz Law in ftt, nevertheless we can prove several cases of it.

5.1.5.1. = is an equivalence relation.

$$\dfrac{a=b \qquad \begin{array}{c} x \in A \\ \vdots \\ x=x \end{array}}{b=a}$$

$$\dfrac{a=b,b=c \qquad \begin{array}{c} x,y \in A \\ \vdots \\ x=y \text{ conv } x=y \end{array}}{\dfrac{b=b \text{ conv } a=c \qquad b=b}{a=c}}$$

5.1.5.2. equational substitutivety

$$\dfrac{x=y}{b[x] = b[y]}$$

5.1.5.3. 1-1 ness of disjunctive injections

$$\dfrac{x,y \in A \qquad i[x] = i[y]}{x=y}$$

use +-elimination to define for $u \in A + B$, $f[u] \in A$

$$f[i[v]] =_{df} i[v]$$

$$f[j[w]] =_{df} x$$

and then apply 5.1.5.2.

5.1.5.4. trichotomy on N

$$\dfrac{x,y \in N}{x=y + \Sigma z \in N.(x+z)=y \;\&\; succ[z] + \Sigma z \in N.(y+z)=x \;\&\; succ[z]}$$

where $succ[z] =_{df} \Sigma t \in N.\ z=(t+1)$

5.1.5.5. Strict decidability of identity with 0.

$$\frac{x \in N}{x=0 \ + \ \mathrm{succ}[x]} \qquad\qquad \frac{x=0 \ \& \ \mathrm{succ}[x]}{0 = 1}$$

(which is the nearest we can approach $x=0 \ + \ x\neq 0$ in ftt).

5.1.5.6. Strict decidability of identity on N

$$\frac{x=y \ \& \ \Sigma z \in N.(z+z)=y \ \& \ \mathrm{succ}[z]}{0=1} \qquad\qquad \frac{x=y \ \& \ \Sigma z \in N.x=(y+z) \ \& \ \mathrm{succ}[z]}{0=1}$$

(which together with 5.1.5.4. is the nearest we can approach $x=y \ + \ x\neq y$ in ftt).

5.1.6. Some meta theorems of ftt

Each of the theorems that follow can be made locally in ftt itself. The first is the statement that every type of ftt is countable, where the ftt formalization of countability of a type A, is the three rules:

$$\frac{x \in A}{e_A[x] \in N} \qquad \frac{n \in N}{d_A[n] \in A+T} \qquad \frac{y \in A}{c_A[y] \in d_A[e_n[y]] =_{df} i[y]}$$

If these rules are derivable in ftt, then we say that A is ftt-countable.

5.1.6.1 Theorem. Every type in ftt is ftt-countable

The method of proof is given on page 109(a).

Group C

If $x \in N$ then define $e_N[x]$ by: $\quad e_N[x] =_{df} x \in N$

If $n \in N$ then define $d_N[n]$ by: $\quad d_N[n] =_{df} i[n] \in N+T$

Then for $y \in N$: $\quad i[y] = i[y]$

$$d_N[y] = i[y]$$

$$d_N[e_N[y]] = i[y]$$

and so define $c_N[y]$ by: $\quad c_N[y] =_{df} r[i[y]]$

Define the e,d and c by induction over the definition of ' is a typesymbol' . And in addition for function constants B of index/value strings V_c / V_c define a function constant F_{e_B} of index/value typesymbols K/H such that

$$F_{e_B}[e_A[n]] = e_{B[A]}[n]$$

and similarly for F_{d_B} and F_{c_B}.

Group D

Assume that we have defined e_A, d_A and for each $x \in A$, $e_{B[x]}$ and $d_{B[x]}$; then define $e_{\Sigma x \in A.B[x]}$ by:

$$e_{\Sigma x \in A.B[x]}[(a,b)] =_{df} [e_A[a], e_{B[x]}[b]] \in N$$

where $[,], ()_o, ()_1$ are as in 2.4.1.

Define $d_{\Sigma(A,B)}$ by first defining terms b,a: if $u \in A$ and $v \in B[u]+T$ then define $b[u,v] \in \Sigma x \in A.B[x]+T$ by:

$$b[u,i[y]] =_{df} i[(u,y)]$$

$$b[u,j[z]] =_{df} j[z]$$

and if $w \in A+T$ and $n \in N$ then define $a[n,w] \in \Sigma x \in A.B[x]+T$ by:

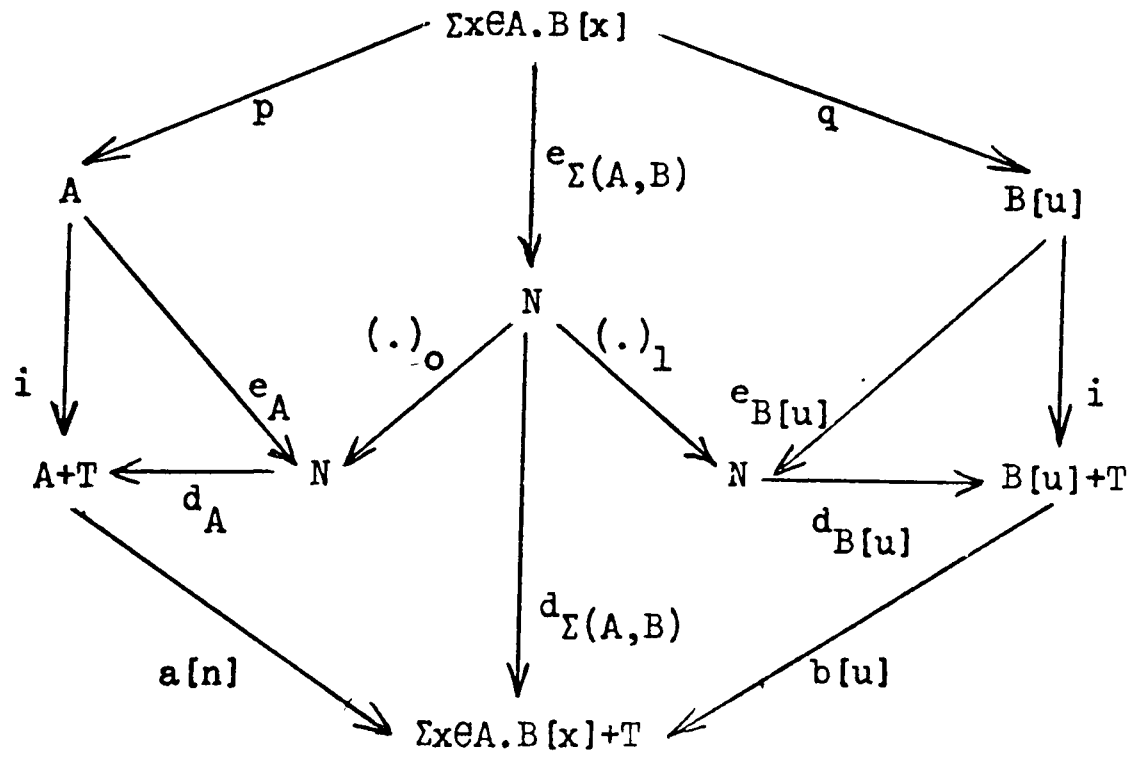
$$a[n,i[y']] =_{df} b[y', d_{B[y']}[(n)_1]]$$

$$a[n,j[z']] =_{df} j[z']$$

Then define $d_{\Sigma(A,B)}$ by:

$$d_{\Sigma(A,B)}[n] =_{df} a[n, d_A[(n)_o]]$$

Pictorially, for $n \in N$ and $u \in A$, we have:



Then for $x \in A$ and $y \in B\,x$:

$$d_{\Sigma(A,B)}[e_{\Sigma(A,B)}[(x,y)]] = a[e_{\Sigma(A,B)}[(x,y)], d_A[(e_{\Sigma(A,B)}[(x,y)])_o]]$$

$$= a[e_{\Sigma(A,B)}[(x,y)], d_A[e_A[x]]]$$

$$= a[e_{\Sigma(A,B)}[(x,y)], i[x]]$$

$$= b[x, d_{B[x]}[(e_{\Sigma(A,B)}[(x,y)])_1]]$$

$$= b[x, d_{B[x]}[e_{B[x]}[y]]]$$

$$= b[x, i[y]]$$

$$= i[(x,y)]$$

hence we can define $c_{\Sigma(A,B)}$ by:

$$c_{\Sigma x \in A.B[x]}[(x,y)] =_{df} r[i[(x,y)]]$$

Group E

If $a,b \in C$ then define $e_{a=b}$ by: $\quad e_{a=b}[z] =_{df} 0 \in N$

To define $d_{a=b}$, first note that:

$$e_c[a] = e_c[b]$$

$$d_c[e_c[a]] = d_c[e_c[b]] \qquad (5.1.5.2.)$$

$$i[a] = i[b] \qquad (5.1.5.3.)$$

$$\overline{\qquad\qquad\qquad\qquad}$$

$$a = b$$

and that by 5.1.5.2., from a=b we have $e_c[a] = e_c[b]$. So by 5.1.5.6. we have the strict decidability of identity on C, and we use this to define $d_{a=b}$:

$$d_{a=b}[0] =_{df} \begin{cases} i[r[a]] & \text{if } a=b \\ j[1^*] & \text{o.w.} \end{cases}$$

$$d_{a=b}[s[c]] =_{df} \quad j[1^*]$$

Then $d_{a=b}[e_{a=b}[z]] = i[r[a]] = i[z]$ by 1.4. . and so define $c_{a=b}$ by:

$$c_{a=b}[z] =_{df} r[i[z]]$$

Group F

Define e_{A+B} by: $\quad e_{A+B}[i[x]] =_{df} [e_A[x]+1,0] \in N$

$$e_{A+B}[j[y]] =_{df} [0,e_B[y]+1] \in N$$

Define d_{A+B} by first defining terms a,b: if $u \in A+T$ define $a[u] \in (A+B)+T$ by:

$$a[i[x]] =_{df} i[i[x]] \in (A+B)+T$$

$$a[j[y]] =_{df} j[y] \quad \in (A+B)+T$$

and if $v \in B+T$ define $b[v] \in (A+B)+T$ by:

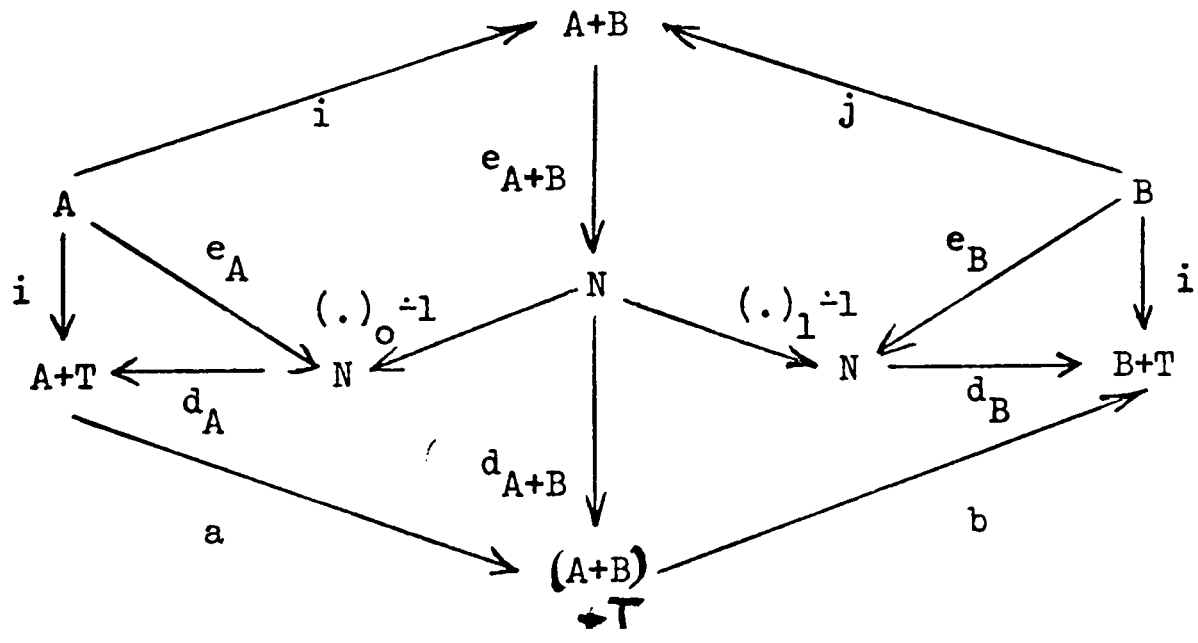
$$b[i[x']] =_{df} i[j[x']] \in (A+B)+T$$

$$b[j[y']] =_{df} j[y'] \quad \in (A+B)+T$$

Then we can define d_{A+B} by:

$$d_{A+B}[n] =_{df} \begin{cases} a[d_A[(n)_o \dot{-} 1]] & \text{if } (n)_1 = 0 \ \& \ \text{succ}[(n)_o] \\ b[d_B[(n)_1 \dot{-} 1]] & \text{if } (n)_o = 0 \ \& \ \text{succ}[(n)_1] \\ j[1^*] & \text{o.w.} \end{cases}$$

Pictorially, we have:



Then for $x \in A$: $d_{A+B}[e_{A+B}[i[x]]] = a[d_A[e_A[x]]]$

$= a[i[x]]$

$= i[i[x]]$

and for $y \in B$: $d_{A+B}[e_{A+B}[j[y]]] = b[d_B[e_B[y]]]$

$= b[i[y]]$

$= i[j[y]]$

hence we can define c_{A+B} by:

$$c_{A+B}[i[x]] =_{df} r[i[i[x]]]$$

$$c_{A+B}[j[y]] =_{df} r[i[j[y]]]$$

Group G

Define e_{N_1} by: $e_{N_1}[x] =_{df} 0 \in N$

Define d_{N_1} by: $d_{N_1}[0] =_{df} i[1^*] \in T+T$

$d_{N_1}[s[c]] =_{df} j[1^*] \in \mathfrak{L}+T$

Then $i[1^*] = i[1^*]$

$d_{N_1}[0] = i[1^*]$

$d_{N_1}[e_{N_1}[1^*]] = i[1^*]$

hence define c_{A+B} by: $c_{A+B}[1^*] =_{df} r[i[1^*]]$

Group H

Where a typesymbol is formed by N,Σ,=,+ or N_1-elimination the definition of e,d and c is given by the induction hypothesis. Thus, for example, if $rec_{A,B[x,Y]}[z]$ is a typesymbol defined by 27) so that

$$rec_{A,B[x,Y]}[0] =_{df} A$$

$$rec_{A,B[x,Y]}[s[c]] =_{df} B[c, rec_{A,B[x,Y]}[c]]$$

$$e_{rec_{A,B}[z]} =_{df} rec_{e_A, F_e}[z]$$

and similarly for d and c. Also

$$e_{un_B[z]} =_{df} un_{e_{B[p(z),q(z)]}}[z]$$

$$e_{id_B[z]} =_{df} id_{e_{B[a]}}[z]$$

$$e_{dis_{F,G}[z]} =_{df} dis_{e_F, e_G}[z]$$

$$e_{r_{i,A}[z]} =_{df} e_A$$

The next thing to note about ftt is that a decidability schema holds of it. That is, for each type A there is a number theoretic function M_A such that:

$$M_A(t)=0 \text{ iff } \Sigma x \in A.\ e_A[x]=t$$

so M_A is a characteristic function of A; and to express this inside ftt we should have to define terms m_A, f_A and g_A such that:

$$\frac{t \in N,\ z \in m_A[t]=0}{f[t,z] \in \Sigma x \in A.\ e_A[x]=t} \qquad \frac{t \in N \qquad y \in \Sigma x \in A.\ e_A[x]=t}{g[t,y] \in m_A[t]=0}$$

We then say that A has an ftt-characteristic term.

5.1.6.2. Theorem. Each typesymbol of ftt has an ftt-characteristic term

Define m_A by induction as on page 109(a), and similarly f_A and g_A. We shall give the definitions of the m's and sketch that of the f's and g's.

Group C

If $t \in N$ then define m by: $m_N[t] =_{df} 0$

If $z \in m_N[t]=0$ then define f_N by: $f_N[t,z] =_{df} (t, r[t])$

If $y \in \Sigma x \in N.\ e_N[x]=t$ then define g_N by: $g_N[t,y] =_{df} r[0]$

Group D

Suppose that we have defined m_A, f_A, g_A and for each $x \in A$, $m_{B[x]}, f_{B[x]}$ and $g_{B[x]}$. Then to define $m_{\Sigma x \in A.B[x]}$ use the decidability of $m_A[n]=0$:

$$m_{\Sigma x \in A.B[x]}[t] =_{df} \begin{cases} m_{B[p[f_A[(t)_0,u]]]}[(t)_1] & \text{if } u \in m_A[(t)_0]=0 \\ 1 & \text{otherwise} \end{cases}$$

Then if $v \in m_{\Sigma x \in A.B[x]}[t]=0$ and $u \in m_A[(t)_0]=0$:

$$(p[f_A[t,u]], p[f_{B[p[f_A[(t)_0,u]]]}[t,v]]) \in \Sigma x \in A.B[x]$$

and $e_{\Sigma x \in A.B[x]}[(p[f_A[t,u]], p[f_{B[p[f_A[(t)_0,u]]]}[t,v]])] = [(t)_0,(t)_1]=t$

and if $v \in m_{\Sigma x \in A.B[x]}[t]=0$ and otherwise then $0=1$.

Then if $w \in \Sigma y \in (\Sigma x \in A.B[x]).e_{\Sigma x \in A.B[x]}[y]=t$:

$$t = [e_A[p[p[w]]], e_{B[p]p[w]]}[q[p[w]]]]$$

and $$g_A[t, p[p[w]]] \in m_A[(t)_0]=0$$

$$g_{B[p[p[w]]]}[t, p[p[w]]] \in m_{B[p[p[w]]]}[(t)_1]=0$$

hence $m_{\Sigma x \in A.B[x]}[t]=0$.

Group E

If $a,b \in C$ then define $m_{a=b}$ by: $$m_{a=b}[t] =_{df} [t]$$

Then by definition of $e_{a=b}$; $m_{a=b}[t] = 0$ iff $\Sigma x \in a=b.\ e_{a=b}[x]=t$

Group F

Define m_{A+B} by:

$$m_{A+B}[t] =_{df} \begin{cases} m_A[(t)_0 \dot{-} 1] & \text{if } (t)_1=0 \ \&\ \text{succ}[(t)_1] \\ m_B[(t)_1 \dot{-} 1] & \text{if } (t)_0=0 \ \&\ \text{succ}[(t)_1] \\ 1 & \text{otherwise} \end{cases}$$

Then if $v \in m_{A+B}[t]=0$, either $m_A[(t)_o \dot{-} 1]=0$ and;

$$f_A[(t)_o \dot{-} 1, v] \in \Sigma x \in A.e_A[x]=(t)_o \dot{-} 1$$

$$e_{A+B}[i[p[f_A[(t)_o \dot{-} 1, v]]]] = [(t)_o, 0] = t$$

or $m_B[(t)_1 \dot{-} 1]=0$ and;

$$f_B[(t)_1 \dot{-} 1, v] \in \Sigma x \in B.e_B[x]=(t)_1 \dot{-} 1$$

$$e_{A+B}[j[p[f_B[(t)_1 \dot{-} 1, v]]]] = [0, (t)_1] = t$$

And if $w \in \Sigma x \in A+B.e_{A+B}[x]=t$, then for $y \in A$, such that $e_{A+B}[i[y]] = t$;

$$e_{A+B}[i[y]] = [e_A[y]+1, 0]$$

and $m_A[e_A[y]] = 0$

$$m_A[(t)_o \dot{-} 1] = 0$$

hence $m_{A+B}[t] = 0$

and for $z \in B$, such that $e_{A+B}[j[z]]=t$;

$$e_{A+B}[j[z]] = [0, e_B[z]+1]$$

and $m_B[e_B[z]] = 0$ by definition

$$m_B[(t)_1 \dot{-} 1] = 0$$

$$m_{A+B}[t] = 0$$

Group G

Define m_{N_1} by: $m_{N_1}[t] = t$

Then $m_{N_1}[t] = 0$ iff $t=e_{N_1}[1^*]$

With regard to the typesymbols formed by Group H the definition of m, f and g is derived from the induction hypothesis and an application of the rules 7),11), 15) or 21) resp. in the same way as 5.1.6.1. group H.

Now, theorem 5.1.6.2. gives us the decidability of every type of ftt, and theorem 5.1.6.1. gives us the decidability of identity. But if we define extensional equality on N_1 and A+B by:

if $x,y \in N_1$ then $x =_{ext} y =_{df} T$

if $z,t \in A+B$ then $z =_{ext} t =_{df} \Sigma x,y \in A.x =_{ext} y \ \& \ z=i[x] \ \& \ t=i[y].$

$$+ \Sigma u,v \in B.u =_{ext} v \ \& \ z=j[u] \ \& \ t=j[v].$$

and col(A) to be the rule:

$$\frac{x,y \in A \qquad x =_{ext} y}{x =_A y}$$

then we can clearly prove that:

5.1.6.3. Theorem. For every type of ftt, $\vdash$ col(A).

§2. The ordinal of ftt

We can establish the proof theoretic ordinal of ftt, by isomorphically embedding it in primitive recursive arithmetic and vice versa. Before defining primitive recursive arithmetic, we define its corresponding calculus of terms, which we call primitive recursive calculus (prc).

5.2.1. The calculus of terms

We begin with a collection of variables: $n_1, n_2, \ldots, m_1, m_2, \ldots$

and a constant: 0'

Then a term of prc is built up from variables and the constant by the following productions:

$$\frac{n}{n+1}$$

$$\frac{\psi(n_1,n_2,\ldots,n_m) \qquad\qquad \chi(n_1,n_2,\ldots,n_m,n_{m+1})}{\phi(n_1,n_2,\ldots,n_m,n_{m+1})}$$

where $\phi(n_1,n_2,\ldots,n_m,0')$ conv $\psi(n_1,n_2,\ldots,n_m)$

$\phi(n_1,n_2,\ldots,n_m,n+1)$ conv $\chi(n_1,\ldots,n_m,0', \phi(n_1,\ldots,n_2,n))$

and the usual rules for conv:

$$a \text{ conv } a \qquad \frac{a \text{ conv } b}{b \text{ conv } a} \qquad \frac{a \text{ conv } b \qquad b \text{ conv } c}{a \text{ conv } c}$$

$$\frac{a_1 \text{ conv } b_1,\ldots,a_n \text{ conv } b_n}{\phi(a_1,a_2,\ldots,a_n) \text{ conv } \phi(b_1,\ldots,b_n)}$$

5.2.2. Some distinguished terms

I assume the definitions of the following terms:

$k(n,m)$ - pairing

$d_1(n),d_2(n)$ - unpairing so that: $d_1(k(n,m))$ conv n

$d_2(k(n,m))$ conv m

$c(n,m)$ - constantly n for each m

$dif(n,m)$ - truncated minus, written: $n \dot{-} m$

$=(n,m)$ - characteristic term of identity, so that: $=(m,m)$ conv 0

$=(m,m+1)$ conv 1

We began with prc, because we can in fact prove:

5.2.3. Theorem. ftt is embeddable in prc

The embedding proceeds as follows. For each closed typesymbol A of ftt assign an open term $\bar{A}(n)$ of prc, and for each closed term $a \in A$, $\bar{a}$

a closed term of prc; in such a way that: $\bar{A}(\bar{a})$ conv 0'.

The assignment for open terms, as usual, requires more. If $b[x_1,\ldots,x_m] \in B[x_1,\ldots,x_m]$ where $x_1 \in A_1,\ldots,x_m \in A_m[x_1,\ldots,x_{m-1}]$ then assign an open term $\bar{b}(n_1,\ldots,n_m)$ of prc and an open term $\bar{B}(n_1,\ldots,n_m,n)$; in such a way that:

for each closed term $a_1,a_2,\ldots,a_m$ if $\bar{A}_1(a_1),\bar{A}_2(a_1,a_2),\ldots\ldots$ $\bar{A}_m(a_1,\ldots,a_{m-1})$ each conv 0' then $\bar{B}(a_1,\ldots,a_m,\bar{b}(a_1,\ldots,a_m))$ conv 0'.

For ease of presentation, we use the simplified formalization, that is we do not consider a free variable throughout the schemata. This will not result in any difficulties as ftt has no variable binding operations.

Also for a non-typesymbol $B[A] \in V_c$, assign a term $\bar{B}(m)$ so that $\overline{B[A]}(n)$ conv $\bar{B}(\bar{A}(n))$.

Group C

i) $\bar{N}(n) =_{df} c(0',n)$

ii) $\bar{0} =_{df} 0'$

$\bar{s}(n) =_{df} n+1$ and write $\bar{s}(0') = 1'$

iii) $\overline{rec_{a,b}}(n) =_{df} \phi(n)$ where

$\phi(0')$ conv $\bar{a}$

$\phi(n+1)$ conv $\bar{b}(n,\overline{rec_{a,b}}(n))$

N-introduction. True by definition.

N-elimination. Assume that $\bar{C}(0',\bar{a})$ conv 0' and that if $\bar{0}(n,m)$ conv 0' then $\bar{C}(n+1,\bar{b}(n,m))$ conv 0'. For a closed term c, such that $\bar{N}(c)$ conv 0', the decidability of conv allows us to argue:

if c conv 0' then

$\bar{C}(c,\overline{rec_{a,b}}(c))$ conv $\bar{C}(0',\bar{a})$ conv 0'

and if c conv d+1 then

$$\bar{C}(c,\overline{rec_{a,b}}(c)) \text{ conv } \bar{C}(d+1,\bar{b}(d,\overline{rec_{a,b}}(d)))$$

and repeat for d, until after at most c steps

$$\bar{C}(c,\overline{rec_{a,b}}(c)) \text{ conv } 0'$$

N-conversion. Follows straight from the definition.

Group D

i) $\Sigma x \in A.B[x]\ (n) =_{df} \psi(n,\bar{A}(d_1(n)))$ where

$$\psi(n,0') \text{ conv } \bar{B}(d_1(n),d_2(n))$$

$$\psi(n,m+1) \text{ conv } \psi(n,m) + 1$$

ii) $\overline{(a,b)} =_{df} k(\bar{a},\bar{b})$

iii) $\overline{un_b}(n) =_{df} \bar{b}(d_1(n),d_2(n))$

Σ-introduction.

Assume that $\bar{A}(a)$ conv 0' and that $\bar{B}(a,b)$ conv 0', then

$$\overline{\Sigma x \in A,B[x]}\ (\overline{(a,b)}) =_{df} \psi((\bar{a},\bar{b}),\bar{A}(d_1(\bar{a},\bar{b})))$$

$$\text{conv } \psi((\bar{a},\bar{b}),\bar{A}(a))$$

$$\text{conv } \psi((\bar{a},\bar{b}),0')$$

$$\text{conv } \bar{B}(d_1((\bar{a},\bar{b})),d_2((\bar{a},\bar{b})))$$

$$\text{conv } \bar{B}(\bar{a},\bar{b}) \text{ conv } 0'$$

Σ-elimination. Assume that if $\bar{A}(n)$ conv 0' and $\bar{B}(n,m)$ conv 0' then $\bar{C}((n,m),\bar{b}(n,m))$ conv 0'. For a closed term c, such that $\overline{\Sigma x \in A.B.[x]}(c)$, conv 0':

if $\bar{A}(d_1(c))$ conv 0' then

$$\overline{\Sigma x \in A.B.[x]}(c) \text{ conv } \psi(c,0')$$

$$\text{conv } \bar{B}(d_1(c),d_2(c))$$

hence $\bar{B}(d_1(c),d_2(c))$ conv 0', so

$$\bar{C}((d_1(c),d_2(c)),\bar{b}(d_1(c),d_2(c))) \text{ conv } 0'$$

and $\bar{C}(c,\overline{un_b}(c))$ conv $\bar{C}(d_1(c),d_2(c)),\bar{b}(d_1(c),d_2(c)))$

conv 0'

and if $\bar{A}(d_1(c))$ conv d+1 then

$$\overline{\Sigma x \epsilon A.B[x]}(c) \text{ conv } \psi(c,d) + 1$$

hence 1' conv 0', so trivially

$$\bar{C}(c,\overline{un_b}(c)) \text{ conv } 0'$$

Σ-conversion is true by definition.

Group E

i) $\overline{a=b}(n) =_{df} \chi(n,=(\bar{a},\bar{b}))$ where

$\chi(n,0')$ conv n

$\chi(n,m+1)$ conv n+1

ii) $\bar{r}(n) =_{df} c(0',n)$

iii) $\overline{id_d}(n)$ $\nu(n)$ where

$\nu(0')$ conv $\bar{d}(\bar{a})$

$\nu(n+1)$ conv $\gamma(n)$ + 1

=-introduction. $\overline{a=a}(\bar{r}(a))$ conv $\chi(0',0')$ conv 0'

=-elimination. Assume that if $\bar{C}(n)$ conv 0' then $\bar{D}(n,n,\bar{r}(n),\bar{d}(n))$ conv 0'. For closed terms c, such that $\overline{a=b}(c)$ conv 0':

if $\bar{a}$ conv $\bar{b}$ then $\overline{a=b}(c)$ conv $\chi(c,0')$ conv c, hence c conv 0', so

$\overline{id_d}(c)$ conv $\bar{d}(\bar{a})$

$\bar{r}(\bar{a})$ conv c

and $\bar{D}(\bar{a},\bar{a},\bar{r}(\bar{a}),\bar{d}(\bar{a}))$ conv 0' - as C(a) conv 0'

hence $\bar{D}(\bar{a},\bar{b},c,\overline{id_d}(c))$ conv 0',

if $\bar{a}$ does not conv $\bar{b}$ then

$$\overline{a=b}(c) \text{ conv } c+1$$

hence 1' conv 0' and so trivially

$$\bar{D}(\bar{a},\bar{b},c,\overline{id_d}(c)) \text{ conv } 0'$$

=-conversion is straightforward.

Group F

i) $\overline{A+B}(n) =_{df} \tau(n,d_2(n))$ where

$$\tau(n,0') \text{ conv } \bar{A}(d_1(n)\dot{-}1)$$

$$\tau(n,m+1) \text{ conv } \sigma(n,d_1(n))$$

and

$$\sigma(n,0') \text{ conv } \bar{B}(d_2(n)\dot{-}1)$$

$$\sigma(n,k+1) \text{ conv } 1'$$

ii) $\bar{i}(n) =_{df} k(n+1,0')$

$\bar{j}(m) =_{df} k(0',m+1)$

iii) $\overline{dis_{e,f}}(n) =_{df} \rho(n,d_2(n))$ where

$$\rho(n,0') \text{ conv } \bar{e}(d_1(n)\dot{-}1)$$

$$\rho(n,m+1) \text{ conv } \bar{f}(m)$$

+-introduction. $\overline{A+B}(\bar{i}(\bar{a}))$ conv $\rho(\bar{i}(\bar{a}),0')$ conv $\bar{A}(\bar{a})$ conv 0' and similarly $\overline{A+B}(\bar{j}(\bar{b}))$ conv 0'.

+-elimination. Assume that if $\bar{A}(n)$ conv 0' then $\bar{C}(\bar{i}(n),\bar{e}(n))$ conv 0' and that if $\bar{B}(m)$ conv 0' then $\bar{C}(\bar{j}(m),\bar{f}(m))$ conv 0'. For a closed term c, such that $\overline{A+B}(c)$ conv 0':

if $d_2(c)$ conv 0' then

$$\overline{A+B}(c) \text{ conv } \tau(c,0') \text{ conv } \bar{A}(d_1(c)\dot{-}1)$$

hence $\bar{A}(d_1(c)-1)$ conv 0', so

$$\bar{C}(\bar{i}(d_1(c)-1),\bar{e}(d_1(c)\dot{-}1)) \text{ conv } 0'$$

but $\bar{C}(\bar{i}(d_1(c)-1),\bar{e}(d_1(c)\dot{-}1))$

$$\text{conv } \bar{C}(k(d_1(c),0'),\bar{e}(d_1(c)\dot{-}1))$$

$$\text{conv } \bar{C}(k(d_1(c),d_2(c)),\bar{e}(d_1(c)\dot{-}1))$$

$$\text{conv } \bar{C}(c,\bar{e}(d_1(c)\dot{-}1))$$

and $\overline{dis}_{e,f}(c) \text{ conv } \bar{e}(d_1(c)\dot{-}1)$

hence $\bar{C}(c,\overline{dis}_{e,f}(c)) \text{ conv } \bar{C}(c,\bar{e}(d_1(c)\dot{-}1)) \text{ conv } 0'$

if $d_2(c)$ does not conv 0' and $d_1(c)$ conv 0' then

$$\bar{B}(d_2(c)\dot{-}1) \text{ conv } 0$$

so: $\bar{C}(\bar{j}(d_2(c)\dot{-}1),\bar{f}(d_2(c)\dot{-}1)) \text{ conv } 0'$

$$\overline{dis}_{e,f}(c) \text{ conv } \bar{f}(d_2(c)\dot{-}1)$$

and the result follows in the same way as before.

If neither $d_1(c)$ nor $d_2(c)$ conv 0' then $\overline{A+B}(c)$ conv 1' so 1' conv 0' and result is trivial.

+-conversion. True by definition.

Group G_1

i) $\overline{N_1}(n) =_{df} n$

ii) $\overline{1^*} =_{df} 0'$

iii) $\overline{r_1(a)}(n) =_{df} c(\bar{a},n)$

N_1-introduction. True by definition.

N_1-elimination.

Assume that $\bar{C}(\overline{1^*},\bar{a})$ conv 0'. For closed terms c, such that $\bar{N}_1(c)$ conv 0':

$$c \text{ conv } 0' \text{ conv } 1$$

hence $$\bar{C}(c,\overline{r(a)}(c)) \text{ conv } \bar{C}(c,\bar{a}) \text{ conv } \bar{C}(\overline{1^*},\bar{a})$$

$$\text{conv } 0$$

Group G'

$\overline{r(C)}(n) =_{df} c(1',n)$

So if $\overline{0=1}(c)$ conv 0 then 1' conv 0' and trivially $\bar{C}(\overline{r(C)}(c))$ conv 0'.

Group H

For the other typeforming rules proceed exactly as in the corresponding term forming cases. So that for example:

$\overline{rec_{A,B}}(n) =_{df} \phi(n)$ where

$$\phi(0') \text{ conv } \bar{A}$$

$$\phi(n+1) \text{ conv } \bar{B}(n,\overline{rec_{A,B}}(n))$$

5.2.4. Primitive recursive arithmetic (pra)

We now extend the primitive recursive calculus of terms to the theory pra, which we shall prove is embeddable in ftt. In this way we shall have established the equivalence of ftt with a more familiar system.

The language of pra comprises: the terms of prc, and for each pair of terms of prc; t_1,t_2 form the atomic formula $t_1=t_2$. And a well formed formula of pra is defined as follows:

i) an atomic formula is a wff

ii) if A and B are wff's then $A \wedge B$, $A \vee B$ and $A \rightarrow B$ are wff's.

The rules of pra are as follows:

∧-introduction:

$$\frac{A \qquad B}{A \wedge B}$$

∧-elimination:

$$\frac{A \wedge B}{A} \qquad \frac{A \wedge B}{B}$$

∨-introduction:

$$\frac{A}{A \vee B} \qquad \frac{B}{A \vee B}$$

∨-elimination:

$$\frac{A \vee B \qquad \begin{matrix} A \\ \vdots \\ D \end{matrix} \qquad \begin{matrix} B \\ \vdots \\ D \end{matrix}}{D}$$

→-introduction:

$$\frac{\begin{matrix} C \\ \vdots \\ D \end{matrix}}{C \to D}$$

→-elimination:

$$\frac{C \qquad C \to D}{D}$$

=-elimination:

$$\frac{x=y \qquad A(x)}{A(y)}$$

induction:

$$\frac{A(0') \qquad \begin{matrix} A(x) \\ \vdots \\ A(x+1) \end{matrix}}{A(x)}$$

Also another rule of conversion:

$$\frac{A \qquad A \text{ conv } B}{B}$$

The axioms: $x=x$

And the definition: $A =_{df} A \to 0'=1'$

The system as we have described it is equivalent to the usual Intuitionistic Primitive Recursive Arithmetic (IPRA) which is defined as follows. Extend the terms of prc by adding the axioms of Peano, the axiom $0'=0'$, the rule of induction and substitution rules for identity. The connectives are defined in truth-functionally.

In IPRA it can be shown that identity is provably decidable and if CPRA is IPRA with the rule of double negation added, then if A is provable in IPRA, it is provable in CPRA.

5.2.5. Theorem. pra is embeddable in prc.

We establish the embedding by assigning to each predicate $A(x_1,\ldots,x_m)$ of pra, a term $a(n_1,\ldots,n_m)$ of prc in such a way that:

$A(x_1,\ldots,x_m)$ iff $a(a_1,\ldots,a_m)$ conv 0', for every string of closed terms of prc; $a_1,\ldots,a_m$

Firstly define the terms cap(n,m), cup(n,m), arr(n,m) and eq(n,m) of prc by the schemata:

cap(0',0') conv 0'	cup(0',0') conv 0'
cap(0',r+1) conv 1'	cup(0',r+1) conv 0'
cap(k+1,0') conv 1'	cup(k+1,0') conv 0'
cap(k+1,r+1) conv cap (k,r)	cup(k+1,r+1) conv cup(k,r)
arr(0',0') conv 0'	eq(0',0') conv 0'
arr(0',r+1) conv 1'	eq(0',r+1) conv 1'
arr(k+1,0') conv 0'	eq(k+1,0') conv 1'
arr(k+1,r+1) conv k	eq(k+1,r+1) conv eq(k,r)

And note that they have the desired properties:

cap(a,b) conv 0' iff a conv 0' or b conv 0'

cup(a,b) conv 0' iff a conv 0' and b conv 0'

arr(a,b) conv 0' iff a conv 1' or b conv 0'

eq(a,b) conv 0' iff a conv b

Then to x=y assign eq(n,m) and if we have assigned $a(n_1,\ldots,n_m)$ to $A(x_1,\ldots,x_m)$ and $b(n_1,\ldots,n_m)$ to $B(x_1,\ldots,x_m)$ then assign to:

$A(x_1,\ldots,x_m) \wedge B(x_1,\ldots,x_m)$; $cap(a(n_1,\ldots,n_m),b(n_1,\ldots,n_m))$

$A(x_1,\ldots,x_m) \vee B(x_1,\ldots,x_m)$; $cup(a(n_1,\ldots,n_m),b(n_1,\ldots,n_m))$

$A(x_1,\ldots,x_m) \to B(x_1,\ldots,x_m)$; $arr(a(n_1,\ldots,n_m),b(n_1,\ldots,n_m))$

It is then straightforward to check that the rules are preserved, and we shall omit the proof here.

5.2.6. Theorem. prc is embeddable in ftt.

The embedding is as follows: to each variable n of prc assign a variable $x_n \in N$; then to a term $a(n_1,\ldots,n_m)$ of prc assign a term $\bar{a}[x_{n_1},\ldots,x_{n_m}] \in N$.

$\overline{0'} =_{df} 0$

$\overline{n+1} =_{df} s[x_n]$

$\overline{\phi(n_1,\ldots,n_m)} =_{df} rec_{\psi,\chi}[x_{n_1},\ldots,x_{n_m},x]$ where

$$rec_{\psi,\chi}[x_{n_1},\ldots,x_{n_m},0] =_{df} \bar{\psi}[x_{n_1},\ldots,x_{n_m}]$$

$$rec_{\psi,\chi}[x_{n_1},\ldots,x_{n_m},s[c]] =_{df} \bar{\chi}[x_{n_1},\ldots,x_{n_m},0,\bar{\phi}[x_{n_1},\ldots,x_{n_m},c]]$$

5.2.7. Corollary. The ordinal of ftt is ω^ω

We firstly make conservative extensions of ftt and pra; ftt* and pra* respectively; as follows. Add to ftt the typesymbol $0 \in V_0$, the term $1 \in 0$, the function constant $ of index/value typesymbols

$$\upsilon/0$$

the rule that if for $x_1 \in A_1, \ldots\ldots, x_n \in A_n[x_1,\ldots,x_{n-1}], x \in N$

$b[x_1,\ldots\ldots,x_n,x] \in \upsilon$

then introduce the function constant $\bigwedge(b[x_1,\ldots,x_n,x])$ with index/value strings

$$A_1,\ldots\ldots,A_n[x_1,\ldots,x_{n-1}]/0$$

and variables of type 0.

Add to ftt the following:

ordinal variables- $\alpha_1, \alpha_2, \ldots$

ordinal terms- $\mathbb{1}$, $\$(\alpha)$

ordinal productions-

i) $$\frac{\psi(n_1, \ldots, n_m, \alpha_1, \ldots, \alpha_k) \qquad \chi(n_1, \ldots, n_m, \alpha_1, \ldots, \alpha_k, n_{m+1})}{\phi(n_1, \ldots, n_m, n_{m+1}, \alpha_1, \ldots, \alpha_k)}$$

where $\phi(n_1, \ldots, n_m, 0', \alpha_1, \ldots, \alpha_k)$ conv $\psi(n_1, \ldots, n_m, \alpha_1, \ldots, \alpha_k)$

$\phi(n_1, \ldots, n_m, n+1, \alpha_1, \ldots, \alpha_k)$ conv $\chi(n_1, \ldots, n_m, 0', \phi(n_1, \ldots, n, \ldots, \alpha_k), \ldots, \alpha_k)$

ii) $$\frac{\psi(n_1, \ldots, n_m, \alpha_1, \ldots, \alpha_k, n)}{\bigwedge_n \psi(n_1, \ldots, n_m, \alpha_1, \ldots, \alpha_k, n)}$$

Now, assign to each term a of ftt , an ordinal term $\bar{\bar{a}}$ of pra* $\bar{\bar{a}}$ is to be a replica of $\bar{a}$. For example

$\overline{\overline{rec_{a,b}}}(n) =_{df} \phi(n)$ where

$$\phi(0') \text{ conv } \bar{\bar{a}}$$

$$\phi(n+1) \text{ conv } \bar{\bar{b}}(n, \overline{\overline{rec_{a,b}}}(n))$$

and $\bar{a}$, $\bar{b}(n,n)$ are ordinal terms.

Then for each closed normal term $a \in O$, assign $|a|$, an ordinal term of pra*.

Case 1 . a=1

$$|1| =_{df} \mathbb{1}$$

case 2. a=$(b)

$$|\$(b)| =_{df} \$(|b|)$$

case 3. a=$b[x]

For all $x \in N$, $b[x] \in O$. Choose a $b^*[x] \in O$ so that $b^*[x]$ denotes the same ordinal as $b[x]$ and nowhere in the derivation of $b^*[x] \in O$ is there a subderivation

$$\begin{array}{c} \vdots \\ 0 \\ \vdots \\ B \\ \vdots \\ 0 \\ \vdots \end{array}$$

where B is at typesymbol other than 0. This is always possible as there is no induction on 0.

It follows that $b^*[x]$ is of the form $(f;a_1[x],\ldots,a_m[x],x)$ where f is a function constant of index/value typesymbols

$$N, A_1[x], \ldots, A_M[x] \;/\; 0$$

Hence

$$|\wedge b[x]| =_{df} \mathbb{M}\, \bar{\bar{f}}(\bar{a}_1(n),\ldots,\bar{a}_m(n),n)$$

By this process we have assigned , for each term of ftt* which denotes an ordinal, a term of pra* which denotes the same ordinal. We can conclude that the ordinal of ftt* and so of ftt is $\leq$ that of pra* or pra. The other way arround is trivial, and as we already know that the ordinal of pra is ω^ω, so is that of ftt.

§3. The ordinal of M-L

The method used by Aczel [1974] to determine the ordinal of Martin-Löf type theory with one universe is to embed it in ramified analysis of level two. This embedding is otherwise useful in that it provides an interpretation of the type theory in a more familiar system. It would be all the more useful if the embedding could be made directly in ramified analysis rather than through Σ^1_1-AC, as done in [1974]. One approach would be to reproduce the number realizability model in ramified analysis. However, I shall postpone the attempt until another work.

REFERENCES

P. H. G. Aczel

1974 'The strength of Martin-Löf's theory of Types with one Universe.' Privately circulated notes.

E. Bishop

1967 Foundations of Constructive Analysis. McGraw-Hill.

J. C. Cole

1971 'Categories of Sets and Models of Set Theory', in Proceedings of the Bertrand Russel Memorial Logic Conference, Denmark. pp. 351-399.

J. Y. Girard

1972 'Interprétation fonctionelle et élimination des coupures de l'arithemétique d'ordre supérieur', Thesis, University of Paris.

P. Hancock

1973 M.Sc.dissertation, University of Oxford.

A. Heiting

1956 Intuitionism, an introduction. North-Holland.

W. A. Howard

1973 'Heriditary majorizable functionals of finite type', in Troelstra [1973].

S. C. Kleene

1952 Introduction to Metamathematics. North-Holland.

1969 'Formalized recursive functionals and formalized realizability'. Memoirs of the American Mathematical Society, no. 89.

S. C. Kleene & R. E. Vesley

1965 The Foundations of Intúitionistic Mathematics. North-Holland.

G. Kreisel & W. A. Howard

1966 'Transfinite induction and bar induction of types zero and one, and the role of continuity in intuitionistic analysis', Journal of Symbolic Logic. pp. 325-359.

S. MacLane

1971 Categories for the working mathematician. Springer-Verlag.

P. Martin-Löf

1972a 'An Intuitionistic Theory of Types'. Privately circulated notes.

1972b 'About models for Intuitionistic Type Theories and the notion of definitional equality'. Report 4, Mathematics Institute, University of Stockholm.

1974 'An Intuitionistic Theory of Types: Predicative Part', Mathematics Institute notes no. 3, Stockholm University.

D. S. Scott

1973 'Lambda Calculus and Data Types'. To appear in Proceedings of the Logic Colloqueum, Bristol.

C. A. Smorynski

1973 'Applications of Kripke Models', in Troelstra [1973].

W. W. Tait

1968 'Constructive Reasoning', in Logic, Methodology and the Philosophy of Science III, pp. 185-201. North-Holland.

A. S. Troelstra

1971a 'Computability of terms and notions of realizability. Report 71-02, Department of Mathematics, University of Amsterdam.

1971b 'Notions of realizability for intuitionistic arithmetic and intuitionistic arithmetic in all finite types', Oslo Proceedings, pp. 369-405.

1973 (editor) 'Metamathematical investigation of intuitionistic arithmetic and analysis'. Lecture notes in mathematics, 344. Springer-Verlag.

L. Wittgenstein

1921 Tractatus Logico-Philosophicus. Routledge & Keagan Paul.

J. I. Zucker

1971 'Proof theoretic studies of systems of iterated inductive definitions and subsystems of analysis'. Thesis, Stanford University.