Numeration Models of λ-calculus

by

Akira Kanda
Department of Computer Science
University of British Columbia

Technical Report 83-3

April 1983

ABSTRACT

Models of λ-calculus have been studied by Church [?] and Scott [7]. In these studies, finding solutions to the isomorphic equations $S \cong [S \to S]$ where $[S \to S]$ is a certain set of functions from S to S is the main issue. In this paper, we present an example of such solutions which fails to be a model of λ-calculus. This example indicates the necessity of careful consideration of the syntax of λ-calculus, especially for the study of constructive models of λ-calculus. Taking this into account, we axiomatically show when a numeration of Ersov [3] forms a model of λ-calculus. This serves as a general framework for countable models of λ-calculus. Various examples of such numerations are studied. An algebraic characterization of this class of numerations is also given.

§1. λ-calculus

The λ-calculus developed by Church [2] is a formal system designed to study equivalence of functions composed from other functions in certain primitive ways. In this section, we briefly overview this calculus. As mentioned in the abstract, since the syntax of λ-calculus plays an important role in the construction of numeration models, we present the syntax in quite a detailed manner.

We start with assuming a countable set V of variables.

Definition 1.1 (λ-terms)

1. If x is a variable in V, then x is a λ-term.
1. If M and N are λ-terms then so is (MN).
3. If x is a variable and M is a λ-term then so is (λx.M) □

Definition 1.2 (Occurrence)

We define a binary relation occurs over λ-terms as follows:

1. X occurs in X.
2. If X occurs in M or in N, then X occurs in (MN).
3. If X occurs in M, then for every variable y, X occurs in (λy.M). □

We write $X \prec Y$ for "X occurs in Y".

Definition 1.3 (Free and bound variables)

An occurrence of a variable x in M is bound if it is inside a part of M of the form λx.M, otherwise it is free. We say x is free in Y if it has a free occurrence in Y. □

Definition 1.4 (Substitution)

For any terms M, N and any variable x, the result M[x:=N] of substituting N for each free occurrence of x in M (and changing bound variables to avoid clashes) is defined as follows:

1. $x[x:=N] \equiv N$
2. $Z[x:=N] \equiv Z$ for all variables $Z \neq x$
3. $(M_1M_2)[x:=N] \equiv ((M_1[x:=N])(M_2[x:=N]))$
4. $(\lambda x.Y)[x:=N] \equiv (\lambda x.Y)$
5. $(\lambda y.Y)[x:=N] \equiv (\lambda y.Y[x:=N])$ if $y \equiv x$ and $y \not\prec N$ or $x \not\prec N$

   $(\lambda z.(Y[z:=y])[x:=N])$ if $y \not\equiv x$ and $y \prec N$ and $x \prec Y$. □

In λ-calculus, each λ-term is considered to be a representation for a function and the following three reduction rules are to establish equivalence of functions:

Reduction Rules

(α): $(\lambda x.M) \to (\lambda y.M[x:=y])$ x is not bound in M and $y \not\prec M$

(β): $((\lambda x.M)N) \to M[x:=N]$

(η): $(\lambda x.Mx) \to M$ $\quad x \not\prec M$ □

§2. Reflexive sets and models of λ-calculus

A reflexive set is a set which is a solution of the following isomorphic equation: $S \cong [S \to S]$ where $[S \to S]$ is a set of functions from S to S.

Given a reflexive set S with an isomorphism pair $(\Phi : S \to [S \to S], \Psi : [S \to S] \to S)$, we build a model of λ-calculus as follows:

--An environment is a function $\rho : V \to S$. We denote the set of all environments by Env. Updating an environment is an operation $Upd : Env \times V \times S \to Env$ such that:

$Upd(\rho,x,s)(v) :=$ if $x = v$ then s else $\rho(v)$.

We write $\rho[x:=s]$ for $Upd(\rho,x,s)$.

--An interpretation is a function $\xi : T \times Env \to S$ defined by:

$\xi(x,\rho) = \rho(x)$

$$\xi((MN),\rho) = \Phi(\xi(M,\rho))(\xi(N,\rho))$$

$$\xi((\lambda x.M),\rho) = \Psi(\lambda s\epsilon S.\xi(M,\rho[x:=s]))$$

It is important to notice that ξ is well-defined only when $\lambda s\epsilon S.\xi(M,\rho[x:=s]) \in [S\to S]$. If this condition is satisfied we say S is an <u>admissible reflexive set</u>.

--Now we can establish the following theorem which states that every admissible reflexive set forms a model of λ-calculus.

<u>Theorem 2.1</u>

(1) $\xi((\lambda x.M),\rho) = \xi((\lambda y.M[x:=y]),\rho)$ provided that x is not bound in M and $y \notin M$.

(2) $\xi(((\lambda x.M)N),\rho) = \xi(M[x:=N],\rho)$

(3) $\xi((\lambda x.Mx),\rho) = \xi(M,\rho)$ provided $x \notin M$.

<u>Proof</u>

(1) $\xi((\lambda y.M[x:=y]),\rho)$

$= \Psi(\lambda s\epsilon S.\xi(M[x:=y],\rho[y:=s]))$

$= \Psi(\lambda s\epsilon S.\xi(M,\rho[y:=s][x:=\xi(y,\rho[y:=s])]))$

$= \Psi(\lambda s\epsilon S.\xi(M,\rho[y:=s][x:=s]))$

$= \Psi(\lambda s\epsilon S.\xi(M,\rho[x:=s]))$ $(\because y \notin M)$

$= \xi((\lambda x.M),\rho)$

(2) $\xi(((\lambda x.M)N),\rho)$

$= \Phi(\xi((\lambda x.M),\rho))(\xi(N,\rho))$

$= \Phi(\Psi(\lambda s\epsilon S.\xi(M,\rho[x:=s])))(\xi(N,\rho))$

$= (\lambda s\epsilon S.\xi(M,\rho[x:=s]))(\xi(N,\rho))$

$= \xi(M,\rho[x:=\xi(N,\rho)])$

$= \xi(M[x:=N],\rho)$

(3) $\xi((\lambda x.Mx),\rho)$

$= \Psi(\lambda s\in S.\xi(Mx,\rho[x:=s]))$

$= \Psi(\lambda s\in S.\Phi(\xi(M,\rho[x:=s]))(\xi(x,\rho[x:=s])))$

$= \Psi(\lambda s\in S.\Phi(\xi(M,\rho[x:=s]))(s))$

$= \Psi(\lambda s\in S.\Phi(\xi(M,\rho))(s))$ $(\because x\notin M)$

$= \Psi(\Phi(\xi(M,\rho))$

$= \xi(M,\rho)$ □

Term model of Church and lattice theoretic model of Scott are examples of admissible reflexive sets. Admissibility is an important part of building models of λ-calculus, which takes care of the inter-relation between syntax and semantics of λ-calculus. In fact, we can easily show examples of reflexive sets which are not models of λ-calculus.

Let [N→N] be the set of all total recursive functions from N to N. Then we have an isomorphism

$$N\cong[N\to N]$$

for we know that [N→N] is a countable set. It is well-known that this isomorphism can never be "effective". Due to this non-constructiveness,

$$\lambda n\in N.\xi(M,\rho[x:=n])$$

fails to be in [N→N]. Thus the reflexive set $N\cong[N\to N]$ does not form a model of λ-calculus.

This example indicates that finding reflexive sets is not difficult but what is not easy is finding reflexive sets which are admissible. Also this indicates that careful treatment of syntax of λ-calculus is important in building models of λ-calculus, especially when it comes to constructive models of λ-calculus.

It ought to be stressed that, in this section, we have established a general result that every admissible reflexive set forms a model of

λ-calculus. Thus building a model of λ-calculus now amounts to building an admissible reflexive set.

In the rest of the paper, we develop a general framework for building models of λ-calculus with effectiveness constraint.

§3. Gödelized λ-calculus

By Gödel numbering variables and λ-terms, we can realize constructions of λ-calculus as recursive functions. It is easy to establish computable bijections $\nu : N \to V$ and $\tau : N \to T$ where T is the set of all λ-terms. The syntactic constructions of λ-calculus corresponds to the following system of recursive functions:

is-var(n) = true if $\tau(n) \in V$
false otherwise

is-apply(n) = true if $\tau(n)$ = (MN) for some $M, N \in T$
false otherwise

is-abstract(n) = true if $\tau(n) = (\lambda x.M)$ for some $x \in V$ and $M \in T$
false

$\tau(\text{inc}(n)) = \nu(n)$

$\nu(\text{var}(n)) = \tau(n)$ if is-var(n)

is-apply(n) => $\tau(\text{apply}(\text{rator}(n), \text{rand}(n)) = \tau(n)$

is-abst(n) $\Rightarrow$ $\tau(\text{abst}(\text{bound}(n), \text{body}(n))) = \tau(n)$

By taking advantage of constructive bijections ν, τ we can identify variables and λ-terms with uniquely corresponding natural numbers. Thus we can abstractly define syntax of λ-terms as the system of recursive functions satisfying:

is-apply(n) => apply(rator(n),rand(n)) = n

is-abstract(n) => abst(bound(n),body(n)) = n

is-var(n) => inc(var(n)) = n

var(in((n)) = n

without referring to ν and τ.

Proposition 3.1

There are recursive functions occur, free, bound, and subst satisfying:

occur(n,m) = true if $\nu(n) \prec \tau(m)$

false otherwise

free(n,m) = true if $\nu(n)$ occurs free in $\tau(m)$

false otherwise

bound(n,m) = true if $\nu(n)$ occurs bound in $\tau(m)$

false otherwise

$\tau(\text{subst}(m,x,n)) = \tau(m)[\nu(x):=\tau(n)]$ □

Conversion rules of λ-calculus are realizable in the following sense:

Proposition 3.2

There are recursive functions $\alpha,\beta.\eta$ s.t.

(1) $\alpha(n,m)$ = true if $\tau(n) \overset{\alpha}{\Rightarrow} \tau(m)$

false otherwise

(2) $\beta(n,m)$ = true if $\tau(n) \overset{\beta}{\Rightarrow} \tau(m)$

false otherwise

(3) $\eta(n,m)$ = true if $\tau(n) \overset{\eta}{\Rightarrow} \tau(m)$

false otherwise □

This almost tedious section is needed to study how constructive syntax of λ-calculus interacts with a constructive reflexive set to form a constructive model of λ-calculus.

§4. Numerated Reflexive Sets

In §2, we generalized (abstracted) various mathematical structures which satisfy the equation $S \cong [S \to S]$ as reflexive sets and studied how reflexive sets form models of λ-calculus. In this section we make abstraction of constructive sets which satisfy $S \cong [S \to S]$ as numerated reflexive sets and in the next section, we study how a numerated reflexive set forms a model of λ-calculus.

Definition 4.1 (Numeration) (Ersov [3])

A numeration γ is a pair (μ, S) where μ is a surjection from N to S. Let $\gamma_1 = (\mu_1, S_1)$ and $\gamma_2 = (\mu_2, S_2)$ be numerations. A morphism from γ_1 to γ_2 is a function $f: S_1 \to S_2$ such that for some recursive function r_f, $f \cdot \mu_1 = \mu_2 \cdot r_f$. Such r_f is called a realization of f w.r.t. μ_1 and μ_2. If r_f is primitive recursive, we say the morphism f is primitive. □

Proposition 4.2 (Ersov [3])

Numerations and morphisms among them form a category.

If $\mu(n) = \mu(m)$ we write $n =_\mu m$. Obviously $=_\mu$ is an equivalence relation on N.

Definition 4.3

Let $\gamma_1 = (\mu_1, S_1)$ and $\gamma_2 = (\mu_2, S_2)$ be numerations. A numeration $\gamma = (\mu, \mathrm{Hom}(\gamma_1, \gamma_2))$ is admissible wrt γ_1 and γ_2 iff there are recursive functions υ, realize, and numerate s.t.

$\mu(n)(\mu_1(m)) = \mu_2(\upsilon(n,m))$

$r_{\mu(n)} = \phi_{\mathrm{realize}(n)}$

if ϕ_n is a realization of $f: \gamma_1 \to \gamma_2$ then $\mu(\mathrm{numerate}(n)) = f$

$<\phi_i>$ is a Godel numbering of partial recursive functions. If such γ exists, we write $(\gamma_1 \to \gamma_2) = ((\mu_1 \to \mu_2), (S_1 \to S_2))$ to denote it. □

Definition 4.4

A numerated reflexive set (in short NRS) is a numeration $\gamma = (\mu, S)$ s.t.

(1) An admissible numeration $(\gamma \to \gamma)$ exists

(2) $\gamma \cong (\gamma \to \gamma)$ in the category of numerations □

§5. Numeration Models of λ-calculus

From an NRS, we build a model of λ-calculus as follows:

Definition 5.1

Let (ν, V) be the bijection discussed in §2 and let (μ, S) be an NRS. Let $(\Psi : S \to (S \to S),\ \Psi' : (S \to S) \to S)$ be an isomorphism pair. An environment is a primitive morphism from (ν, V) to (μ, S). We write Env to denote the set of all environment. □

Since every λ-term has only finitely many occurrences of variables, it is sufficient to consider only primitive morphisms from (ν, V) to (μ, S).

The next theorem states that updating environments has a realization:

Theorem 5.2

(1) For each $\rho \in Env$, $x \in V$ and $s \in S$, $\rho[x := s] \in Env$.

(2) There is a recursive function $Update : N^3 \to N$ s.t. if $r_\rho = \psi_i$ then

$\psi_{Update(i,n,m)} = r_{\rho[\nu(n) := \mu(m)]}$,

where $<\psi_i>$ is a Gödel numbering of primitive recursive functions. □

We introduce a numeration (σ, Env) by:

$$\sigma_i = \rho \text{ s.t. } r_\rho = \psi_i.$$

As for the nonconstructive case, the interpretation function can be defined by:

$$\xi(\tau(n),\sigma_i) := \underline{\text{if}}\ \text{is-var}(n)\ \underline{\text{then}}\ \sigma_i(\tau(n))$$
$$\underline{\text{else}}\ \underline{\text{if}}\ \text{is-apply}(n)\ \underline{\text{then}}$$
$$\Phi(\xi(\tau(\text{rator}(n),\sigma_i))(\xi(\tau(\text{rand}(n),\sigma_i)))$$
$$\underline{\text{else}}\ \underline{\text{if}}\ \text{it-abst}(n)\ \underline{\text{then}}$$
$$\Psi(\lambda s\in S.\xi(\tau(\text{body}(n)),\sigma_i[\tau(\text{bound}):=s]))$$

The well-definedness of ξ can be established as a part of the proof of the following theorem:

<u>Theorem 5.3</u>

The following recursive function realizes ξ:

$$e(n,i) := \underline{\text{if}}\ \text{is-var}(n)\ \underline{\text{then}}\ \psi_i(\text{var}(n))$$
$$\underline{\text{else}}\ \underline{\text{if}}\ \text{is-apply}(n)\ \underline{\text{then}}$$
$$\cup(r_\Phi\cdot e(\text{vator}(n),i),\ e(\text{rand}(n),i))$$
$$\underline{\text{else}}\ \underline{\text{if}}\ \text{is-abst}(n)\ \underline{\text{then}}$$
$$r_\Psi(\text{numerate}\cdot\text{ab}(n,i))$$

where ab is a recursive function satisfying:

$$\phi_{ab(n,i)}(m) = e(\text{body}(n),\ \text{Update}(i,\text{bound}(n),m)).$$

<u>Proof</u>

By induction on the structure of terms we will show:

$$\mu(e(n,i)) = \xi(\tau(n),\sigma_i)$$

(Base) is-var(n) = <u>true</u>:

$$\mu(e(n,i)) = \mu(\psi_i(\text{var}(n))) = \sigma_i(\tau(n)) = \xi(\tau(n),\sigma_i)$$

(Step) <u>Case 1</u>: is-apply(n) = <u>true</u>:

$$\mu(e(n,i)) = \mu(\cup(r_\Phi\cdot e(\text{rator}(n),i),\ e(\text{rand}(n),i)))$$
$$= (\mu\to\mu)(r_\Phi\cdot e(\text{rator}(n),i))(\mu(e(\text{rand}(n),i)))$$
$$= \Phi(\mu(e(\text{rator}(n),i)))(\mu(e(\text{rand}(n),i))$$

$= \Phi(\xi(\tau(\mathrm{rator}(n)),\sigma_i))(\xi(\tau(\mathrm{rand}(n)),\sigma_i))$ (Induction Hypothesis)

$= \xi(\tau(n),\sigma_i)$

<u>Case 2</u>: is-abst(n) = <u>true</u>

$$\mu(e(n,i)) = \mu(r_\Psi(\mathrm{numerate}(ab(n,i))))$$
$$= \Psi((\mu\to\mu)(\mathrm{numerate}\cdot ab(n,i)))$$

But $\lambda s\in S.\xi(\tau(\mathrm{body}(n)),\sigma_i[\tau(\mathrm{bound}(n)):=s])$ is realized by

$\phi_{ab(n,i)} = \lambda m\in N.e(\mathrm{body}(n), \mathrm{Update}(i,\mathrm{bound}(n),m))$ because we have:

$$\mu(\phi_{ab(n,i)}(k))$$
$$= \mu(e(\mathrm{body}(n), \mathrm{Update}(i,\mathrm{bound}(n),k)))$$
$$= \xi(\tau(\mathrm{body}(n)),\sigma_i[\mathrm{bound}(n):=\mu(k)]) \quad \text{(Induction Hypothesis)}$$
$$= (\lambda s\in S.\xi(\tau(\mathrm{body}(n)),\sigma_i[\tau(\mathrm{bound}(n):=s])(\mu(k)).$$

Thus by the admissibility of $(\mu\to\mu)$ we have:

$$(\mu\to\mu)(\mathrm{numerate}\cdot ab(n,i)) =$$
$$\lambda s\in S.\xi(\tau(\mathrm{body}(n)),\sigma_i[\mathrm{bound}(n):=s]). \quad \text{----- (I)}$$

Thus

$$\mu(e(n,i)) = \Psi(\lambda s\in S.\xi(\tau(\mathrm{body}(n)),\sigma_i[\mathrm{bound}(n):=s]))$$
$$= \xi(\tau(n),\sigma_i). \qquad \square$$

It is important to notice that (I) establishes

$$\lambda s\in S.\xi(\tau(\mathrm{body}(n)),\sigma_i[\mathrm{bound}(n):=s]) \in (S\to S).$$

Therefore the interpretation function ξ is well-defined. This implies that the admissibility of NRS coincides with the admissibility of reflexive sets discussed in §2.

Exactly the same proof as that of theorem 2.1 establishes the following theorem which states that every NRS forms a model of λ-calculus under the interpretation ξ:

Theorem 5.4

(1) If $\alpha(m,n)$ = true then

$$\xi(\tau(n),\sigma_i) = \xi(\tau(m),\sigma_i)$$

(2) If $\beta(m,n)$ = true then

$$\xi(\tau(n),\sigma_i) = \xi(\tau(m),\sigma_i)$$

(3) If $\eta(m,n)$ = true then

$$\xi(\tau(n),\sigma_i) = \xi(\tau(m),\sigma_i) \qquad \square$$

In λ-calculus, we have a fixed-point combinator Y given by:

$$Y \equiv \lambda f.((\lambda x.f(xx))(\lambda x.f(xx))).$$

By β-reduction we have:

$$Y(f) \overset{\beta}{\rightarrow} (\lambda x.f(xx))(\lambda x.f(xx)) \overset{\beta}{\rightarrow} f(\lambda x.f(xx))(\lambda x.f(xx)).$$

Thus $\xi((Yf),\rho) = \xi((f(Yf)),\rho)$

$= \Phi(\xi(f,\rho))(\xi(Yf),\rho))$ for all $\rho \in Env$.

This indicates the following fixed-point theorem of NRS.

Theorem 5.5 (The Fixed-point Theorem)

If $\gamma = (\mu,S)$ is a numerated reflexive set then there is a recursive function fix s.t.

$$(\mu\rightarrow\mu)(n)(\mu(fix(n))) = \mu(fix(n)).$$

Proof

It can readily be seen that the following predicates hold in intuitionistic sense:

$$\forall x \exists n_x \exists i_x [x =_{(\mu\rightarrow\mu)} r_\Phi(e(n_x,i_x))]$$

$$\forall x \exists n_x \exists i_x [x =_\mu e(n_x,i_x)].$$

let $Y = \tau(k)$. Then we have:

$$(\mu\to\mu)(x)(\mu(e(apply(k,n_x),i_x)))$$
$$= \mu(e(apply(k,n_x),i_x)$$

because Y is a closed λ-term. Let fix:N→N be defined by:

$$fix(x) := e(apply(k,n_x),i_x).$$

Then $(\mu\to\mu)(x)(\mu(fix(x))) = \mu(fix(x))$.

Obviously fix is a recursive function. □

Remark We have established that in order to show a numeration forms a model of λ-calculus, it is sufficient to show that the numeration is an NRS. It is important to notice that characterization of NRS does not explicitly refer to the syntax of λ-calculus at all.

§6. Examples of Numerated Reflexive Sets

In this section, we study a few examples of NRS's to make sure that what we studied in the previous section is not vacuous.

(example 1) Our first example is Church's term model. We present a numeration of term models and show it is an NRS.

We start with defining an equivalence relation ~ over T. The reduction rules α,β,η can be considered as binary relations over T. We define ~ as the smallest equivalence relation containing $\alpha\cup\beta\cup\eta$. The domain of interpretation of term model is TM defined by:

$$TM = \{[t] \mid t\in T\}$$

where [t] is the equivalence class of t with respect to ~. A λ-term f∈T defines a function f from TM to TM as follows:

$$\bar{f}([t]) = [(ft)]$$

Let (TM→TM) be the following set of functions from TM to TM:

$$(TM\to TM) = \{\bar{f} \mid f\in T\}.$$

Then $\Phi: TM \to (TM \to TM)$ and $\Psi: (TM \to TM) \to TM$ given by:

$$\Phi([t]) = \bar{t}$$

$$\Psi(\bar{f}) = [f]$$

establishes an isomorphism:

$$TM \simeq (TM \to TM).$$

The function Φ is well-defined because if $t_1 \sim t_2$ then

$$\bar{t}_1([t]) = [(t_1 t)] = [(t_2 t)] = \bar{t}_2([t]).$$

It can readily be seen that TM is admissible.

Now we study a numerated version of term model. Let $\gamma = (\mu, TM)$ be the following numeration of TM:

$$\mu(n) = [\tau(n)].$$

Theorem 6.1

$g \in (TM \to TM)$ iff g is a morphism from γ to γ.

Proof

If $g \in (TM \to TM)$ then $g = \bar{f}$ for some $f \in T$. Let $f = \tau(k)$. Then

$$\begin{aligned} g(\mu(n)) &= \bar{f}([\tau(n)]) \\ &= [(\tau(k)\tau(n))] \\ &= \mu(apply(k,n)) \end{aligned}$$

Let $\gamma_g(n) = apply(k,n)$. Then r_g is recursive and it realizes g. Therefore $g \in Hom(\gamma \to \gamma)$. Now assume $g \in Hom(\gamma, \gamma)$. Then for some recursive function $\gamma_g: N \to N$, $g(\mu(n)) = \mu(r_g(n))$. By the λ-definability theorem of Church, there is a λ-term $\tau(i)$ such that $(\tau(i)\tau(n)) \to \tau(\gamma_g(n))$. Therefore $g = \overline{\tau(i)}$. Thus $f \in (TM \to TM)$. □

This theorem establishes the equality $Hom(\gamma,\gamma) = (TM \to TM)$. Now we introduce a numeration $(\mu\to\mu): N \to Hom(\gamma,\gamma)$ s.t.

$(\mu\to\mu)(m) = \overline{\tau(m)}$.

Theorem 6.2

(1) The numeration $(\gamma\to\gamma) = ((\mu\to\mu), (TM\to TM))$ is admissible.

(2) $\gamma \cong (\gamma\to\gamma)$ in the category of numerations

Proof

(1) $(\mu\to\mu)(m)(\mu(n)) = \overline{\tau(m)}([\tau(n)])$

$= [\tau(apply(m,n))]$

$= \mu(apply(m,n))$

Thus apply is such U.

$r_{(\mu\to\mu)(m)}(x) = apply(m,x)$

$= \phi_{realize(m)}(x)$

where realize is a primitive recursive function due to the S-m-n theorem.

In the proof of theorem 6.1, the construction of $\tau(i)$ from r_g is uniform. Thus a recursive function numerate satisfying:

if ϕ_n realizes $g \in Hom(\gamma,\gamma)$ then $(\mu\to\mu)(numerate(n)) = g$ exists.

Thus $(\gamma\to\gamma)$ is admissible

(2) $\Phi(\mu(n)) = \Phi([\tau(n)]) = \overline{\tau(n)} = (\mu\to\mu)(n)$

$\Psi((\mu\to\mu)(m)) = \Psi(\overline{\tau(m)}) = [\tau(m)] = \mu(m)$

Thus identity function $N\to N$ realizes both Φ and Ψ.

Thus Φ and Ψ are morphisms.

Thus $\gamma \cong (\gamma\to\gamma)$ in the category of numerations. □

This theorem establishes that the numeration γ is an NRS.

(example 2) Now we show an order theoretic NRS. We form an effective reflexive set and directed indexing as an NRS. For details see Kanda [4], [5].

A domain is a partially ordered set $(X,<)$ such that

(1) For every subset $Z \subset X$, if Z has an upper bound then the least upper bound (list) $\bigsqcup Z$ exists

(2) The set B_D of compact elements of X is countable

(3) For every element $x \in X$, $B_x = \{b \in B \mid b < x\}$ is directed and $x = \bigsqcup B_x$.

Let $\varepsilon : N \to B_D$ be a numeration. (X,ε) is an effectively given domain if there is a pair (b,ℓ) of recursive predicates satisfying:

$b(x) \leftrightarrow \varepsilon(f_p(x))$ has upper bound

$\ell(x,k) \leftrightarrow \varepsilon(k) = \bigsqcup \varepsilon(f_p(x))$

where f_p is the standard enumeration of finite subsets of N. An element $x \in X$ is computable w.r.t. ε if for some recursively enumerable set W, $\varepsilon(W)$ is directed and $x = \bigsqcup \varepsilon(W)$. $Comp(x,\varepsilon)$ denotes the set of all computable elements of (X,ε).

For every effectively given domain (X,ε) there is a recursive function $d_\varepsilon : N \to N$ s.t. for every $j \in N$, $E(W_{d_\varepsilon(j)})$ is directed and if $\varepsilon(W_i)$ is directed then $\bigsqcup \varepsilon(W_i) = \bigsqcup \varepsilon(W_{d_\varepsilon(i)})$. This function gives us a numeration $\delta_\varepsilon : N \to Comp(X,\varepsilon)$ defined by:

$\delta_\varepsilon(i) = \bigsqcup \varepsilon(W_{d_\varepsilon(i)})$.

Given effectively given domains (X,ε) and (X',ε'), let $[X \to X']$ be the set of all functions called continuous functions $f : X \to X'$ which preserve the lub of directed subsets, (i.e. if $D \subset X$ is directed then $f(D) = \{f(x) \mid x \in D\}$ is directed and $f(\bigsqcup D) = \bigsqcup f(D)$.) with the following partial ordering: for $f,g \in [X \to X']$,

$f < g$ iff $f(x) < g(x)$ for all $x \in X$.

It is well-known that $[X \to X']$ is a domain where

$B_{[X \to X]}$ = the set of all possible finite joins of the step functions

$[b,b']:X \to X'$ s.t. $b \in B_X$, $b' \in B_X'$ and

$[b,b'](x) :=$ if $b < x$ then b' else $\perp$

where $\perp = \bigsqcup \phi$.

Let $(\varepsilon \to \varepsilon'):N \to B_{[X \to X']}$ be the following numeration:

$(\varepsilon \to \varepsilon')(n) :=$ if $\sigma(n)$ has a lub then $\bigsqcup \sigma(n)$ else

$\sigma(n) = \{[\varepsilon(i),\varepsilon'(j)] \mid (i,j) \in Pr(n)\}$

Pr:standard enumeration of finite subsets of NxN.

It is known that if (X,ε) and (X',ε') are effectively given domains then $([X \to X'],[\varepsilon \to \varepsilon'])$ is also an effectively given domain.

$f:X \to X'$ is <u>computable</u> wrt $(\varepsilon,\varepsilon')$ iff $f \in Comp([X \to X'],[\varepsilon \to \varepsilon'])$.

<u>Proposition 6.3</u> (Weihranch & Schafer [8])

Let (X,ε) and (X',ε') be effectively given domains. $f:X \to X'$ is computable wrt $(\varepsilon,\varepsilon')$ iff $f\lceil Comp(X',\varepsilon')$ is a morphism from $(\delta_\varepsilon, Comp(X,\varepsilon))$ to $(\delta_{\varepsilon'}, Comp(X',\varepsilon'))$. □

This theorem is a generalization of Myhill-Scheferdason theorem.

Now we are ready to build a constructive reflexive set. Let (X_0,ε_0) be any effectively given domain. Define effectively given domain (X_n,ε_n) recursively by:

$$X_{n+1} = [X_n \to X_n]$$
$$\varepsilon_{n+1} = [\varepsilon_n \to \varepsilon_n]$$

Between (X_n,ε_n) and $(X_{n+1},\varepsilon_{n+1})$ there is an obvious projection pair $(f_n:X_n \to X_{n+1},\ f_n^R:X_{n+1} \to X_n)$ defined by:

$$f_0(x) = \lambda y.x, \qquad f_0^R(x) = x(\perp)$$
$$f_{n+1}(x) = f_n \cdot x \cdot f_n^R \qquad f_{n+1}^R(x) = f_n^R \cdot x \cdot f_n \quad .$$

It is obvious that both f_n and f_n^R are computable. It is not difficult to observe that for each $n \in N$, we can uniformly generate a pair (b_n, ℓ_n) of recursive predicates which makes ε_n effectively given, and n_1, n_2 s.t. $f_n = \delta_{[\varepsilon_n \to \varepsilon_{n+1}]}(n_1)$ $f_n^R = \delta_{[\varepsilon_{n+1} \to \varepsilon_n]}(n_2)$. This makes $((X_n, \varepsilon_n), (f_n, f_n^R))$ an effective sequence of embedding of effectively given domains as in [4,5].

The inverse limit $(X_\infty, \varepsilon_\infty)$ is defined by:

$$X_\infty = \{(x_0, x_1, x_2, \ldots) \mid x_n \in X_n, x_n = f_n^R(x_{n+1})\}$$

$$(x_0, x_1, x_2, \ldots) < (y_0, y_1, \ldots) \text{ iff } x_i < y_i \text{ for all } i.$$

$$\varepsilon_\infty((n,m)) = f_{n\infty}(\varepsilon_n(m))$$

where $(f_{n\infty} : X_n \to X_\infty,\ f_{n\infty}^R : X_\infty \to X_n)$ is a projection pair defined by:

$$f_{n\infty}(x) = (f_0^R \cdot f_1^R \cdots \cdot f_{n-1}^R(x), \ldots, f_{n-1}^R(x), x, f_1(x), f_2 \cdot f_1(x), \ldots)$$

$$f_{n\infty}^R((x_0, x_1, \ldots)) = x_n.$$

As shown in [4], $(X_\infty, \varepsilon_\infty)$ is an effectively given domain and there is a computable isomorphism due to Scott [7]:

$$X_\infty \underset{\Psi}{\overset{\Phi}{\rightleftarrows}} [X_\infty \to X_\infty]$$

$$\Phi(x) = \lambda y \in X_\infty . \bigsqcup (x_{n+1}(y_n))$$

$$\Psi(g) = \langle g_{(0)}, g_{(1)}, \ldots \rangle$$

where $x_k = f_{k\infty}^R(x)$

$$g_{(0)} = f_{0\infty}^R(g(\bot))$$

$$g_{(n+1)} = f_{n\infty}^R \cdot g \cdot f_{n\infty}.$$

Think of the following numerations:

$$\gamma = (\delta_{\varepsilon_\infty}, \mathrm{Comp}(X_\infty, \varepsilon_\infty))$$

$$(\gamma \to \gamma) = (\delta_{[\varepsilon_\infty \to \varepsilon_\infty]}, \mathrm{Comp}([X_\infty \to X_\infty], [\varepsilon_\infty \to \varepsilon_\infty])).$$

Theorem 6.4

(1) The numeration $(\gamma \to \gamma)$ is admissible

(2) $\gamma \cong (\gamma \to \gamma)$ in the category of numerations.

Therefore γ is an NRS.

Proof

(1) It can readily be seen that $ap: [X \to X] \times X \to X$ s.t. $ap(f,x) = f(x)$ is computable. Thus there is a recursive function U s.t.

$$\begin{aligned}\delta_{[\varepsilon \to \varepsilon]}(n)(\delta_{\varepsilon}(m)) &= \delta_{[\varepsilon \to \varepsilon] \times \varepsilon}(n,m) \\ &= \delta_{\varepsilon}(U(n,m)).\end{aligned}$$

The existence of recursive functions realize, numerate as in 4.3 is due to the constructiveness of the proof of 6.3 which appeared in [8].

(2) $(\Phi : X_{\infty} \to [X_{\infty} \to X_{\infty}],\ \Psi : [X_{\infty} \to X_{\infty}] \to X_{\infty})$ is a computable isomorphism pair. Thus $\gamma \cong (\gamma \to \gamma)$ in the category of numerations. □

§7. An Algebraic Characterization of Numerated Reflexive Sets

A countable applicative system is an algebra $(X, \cdot)$ where $\cdot$ is a binary operation over a countable set X. The set T(X) of terms (using countably many variables $x_0, x_1, \ldots$) over $(X, \cdot)$ is inductively defined as follows:

$$x_i \in T(X)$$

$$a \in X \longrightarrow a \in T(X)$$

$$A, B \in T(X) \longrightarrow (A \cdot B) \in T(X).$$

We assume that $\cdot$ associates to the left, also we will drop $\cdot$ if it does not cause much difficulty to read. Thus $A_1 A_2 \ldots A_n$ denotes $(\cdots(A_1 A_2)\cdots A_n)$. To denote that $A \in T(X)$ has variables $x_0, x_1, \ldots, x_n$, we write $A(x_0, x_1, \ldots, x_n)$. We can Godel number terms. Let $\sigma: N \to T(X)$ be a Gödel numbering of terms.

Let $\gamma = (\mu : N \to X, X)$ be a numeration. We define $\gamma \times \gamma$ to be a numeration $(\mu \times \mu : N \to X \times X, X \times X)$ s.t.

$$\mu \times \mu(\langle n,m \rangle) = (\mu(n), \mu(m))$$

where $\langle -,- \rangle$ is a pairing function.

Definition 7.1

A realizably extensional combinatory algebra (RECA) is a 4-tuple $(X, \cdot, \gamma, \sigma)$ s.t.

(1) $(X, \cdot)$ is a countable applicative system

(2) $\gamma = (\mu : N \to X, X)$ is a numeration

(3) $\cdot : X \times X \to X$ is a morphism from $\gamma \times \gamma$ to γ, realized by a recursive function op.

(4) There is a recursive function λ s.t. if $\sigma(n) = A(x_1, \ldots, x_n)$ then $\mu(\lambda(n)) = f$ is a unique element of X satisfying:

$$f y_1 \ldots y_n = A(x_1 := y_1, \ldots, x_n := y_n)$$

where $A(x_1 := y_1, \ldots, x_n := y_n)$ is the result of substituting y_i for x_i in A $(1 \le i \le n)$. □

Note that (4) is a realization of extensional combinatory completeness of Church [1]. Thus an RECA is a countable applicative system $(X, \cdot)$ where $\cdot$ is realizable and the extensional combinatory completeness is also realizable.

Definition 7.2

An RECA $(X, \cdot, \gamma, \sigma)$ is computationally complete iff there is a recursive function alg s.t. if ϕ_n realizes $f : X \to X$ then $\sigma(\text{alg}(n))$ is a term with a free variable, say x and

$$f(z) = (\sigma(\text{alg}(n)))(x := z). \quad \square$$

Lemma 7.3

If $(X,\cdot,\gamma,\sigma)$ is a computationally complete RECA then $Hom(\gamma,\gamma) = \Phi(X)$ where Φ maps elements of X to functions $X \to X$ defined by:

$$\Phi(x)(x') = x \cdot x'.$$

Proof

$f \in Hom(\gamma,\gamma)$ implies some recursive function ϕ_n realizes f. Thus $\sigma(alg(n))$ is a term with a variable, say x s.t.:

$$f(z) = \sigma(alg(n))(x:=z).$$

By realizable extensional completeness,

$$\sigma(alg(n))(x:=z) = \mu(\lambda \cdot alg(n)) \cdot z.$$

Thus $\Phi(\mu(\lambda \cdot alg(n))) = f$. Thus $Hom(\gamma,\gamma) \subset \Phi(X)$. The converse is due to the realizability of the operation. □

Now let $(\gamma \to \gamma) = (\mu \to \mu, Hom(\gamma,\gamma))$ be the following numeration:

$$(\mu \to \mu)(n) = \Phi(\mu(n)).$$

Lemma 7.4

$(\gamma \to \gamma)$ is admissible

Proof

$$\begin{aligned}((\mu \to \mu)(n))(\mu(m)) &= \Phi(\mu(n))(\mu(m)) \\ &= \mu(n) \cdot \mu(m) \\ &= \mu(op(n,m)) \cdot\end{aligned}$$

Thus op serves as $\cup$ of definition 4.3 Let $r_{(\mu \to \mu)(n)}$ be a recursive function s.t.

$$r_{(\mu\to\mu)(n)}(m) = op(n,m).$$

By S-m-n theorem, there is a recursive function realize s.t.

$$\phi_{realize(n)} = r_{(\mu\to\mu)(n)}.$$

But $((\mu\to\mu)(n))(\mu(m)) = \mu(op(n,m)) = \mu(r_{(\mu\to\mu)(n)}(m))$.

By the proof of 7.3, if ϕ_n realizes $f:X\to X$,

$$f = \Phi(\mu(\lambda\cdot alg(n))) = (\mu\to\mu)(\lambda\cdot alg(n)).$$

Thus $\lambda\cdot alg$ serves as numerate of 4.3. □

Theorem 7.5

If $(X,\cdot,\gamma,\sigma)$ is a computationally complete RECA, then γ is a NRS.

Proof

It is sufficient to show $\gamma\cong(\gamma\to\gamma)$ in the category of numerations.

Define $\Psi:Hom(\gamma,\gamma)\to X$ by:

$$\Psi(\Phi(x)) = x.$$

Then $\Phi(\Psi(\Phi(x))) = \Phi(x)$. Thus (Φ,Ψ) is an isomorphism. Both Φ and Ψ are realized by the identity function $N\to N$. Thus $\gamma\cong(\gamma\to\gamma)$ in the category of numerations. □

Before we prove the converse of 7.5, notice that we did not include constants in our language of λ-calculus. By trivial modification of 1.1 and definition of ξ, we can include constants. Under this modification, obviously 5.3, 5.4, 5.5 still hold.

Now let $\gamma = (\mu,X)$ be a NRS with an isomorphism pair (Φ,Ψ). Define an operation $\cdot:X\times X\to X$ by:

$$x\cdot y = \Phi(x)(y).$$

Theorem 7.6

If $\gamma = (\mu, X)$ is a NRS then $(X, \cdot, \gamma, \sigma)$ is a computationally complete RECA.

Proof

(1) Since Φ and $\Phi(x)$ are morphisms, $\cdot$ is a morphism from $\gamma \times \gamma$ to γ.

(2) Let $\sigma(n) = A(x_1,\ldots,x_k)$. Then $\xi(\lambda x_1 .. \lambda x_k . A(x_1,\ldots x_k), \rho) = f$ is the unique elememt of X s.t. $f y_1 .. y_k = A(x_1 := y_1, \ldots, x_k := y_k)$. By 5.3 a recursive function λ s.t. $\mu(\lambda(n)) = f$ exists.

(3) Assume a recursive function ϕ_n realizes $f: X \to X$. Then for some recursive function g, $\Psi(f) = \mu(g(n))$. Let x be a variable and $\mu(g(n)) \cdot x = \sigma(alg(n))$. Then alg is a recursive function and $f(z) = (\sigma(alg(n))(x:=z)$. $\sqcap$

§8. Concluding Remarks

It is possible to think of some interesting sub-category of numerations. For example, we can define a category of continuous numerations.

A continuous numeration is a numeration $\gamma = (\mu, X)$ s.t. X is a partially ordered set and there is a recursive function lim s.t. for each $k \in N$, if $\mu(W_k)$ is directed then $\bigsqcup \mu(W_k)$ exists and $\bigsqcup \mu(W_k) = \mu(\lim(k))$. Let $\gamma = (\mu, X)$ and $\gamma' = (\mu', X')$ be continuous numerations. A continuous morphism from γ to γ' is a morphism $f: \gamma \to \gamma'$ s.t. for every r.e. set W with $\mu(W)$ directed, $f(\bigsqcup \mu(W)) = \bigsqcup f(\mu(W))$.

It can readily be seen that continuous numerations and continuous morphisms form a category.

Same definitions as 4.3 and 4.4 applied for the category of continuous numerations give us a notion of continuous numerated reflexive set (CNRS).

The interpretation function ξ defined in §5 interprets all λ-terms into CNRS. Well definedness of ξ is checked by observing that

$$\lambda s \in S.\xi(\tau(body(n)),\sigma_i[\tau(bound(n):=s])$$

is a continuous morphism form (μ,S) to itself.

Theorem 5.4 holds also for continuous case thus we can conclude that every CNRS forms a model of λ-calculus under the interpretation ξ. We call such models <u>continuous numeration models</u> of λ-calculus.

In fact $(\delta_{\varepsilon_\infty},Comp(X_\infty,\varepsilon_\infty))$ is an example of CNRS. In case theorem 6.3 did not hold, we could still show $(\delta_{\varepsilon_\infty},Comp(X_\infty,\varepsilon_\infty))$ was a model of λ-calculus as a continuous numeration model.

It is an interesting open question if there is any continuous numeration model which is not a numeration model.

Barendregt [1] showed that extensional combinatory systems serve as models of λ-calculus. 7.5 is a numeration version of his result. Meyer [6] characterized models of non-extensional λ-calculus as combinatory systems. It should be a routine modification of our development to obtain numeration version of Meyer's result.

Acknowledgement

This research was motivated by a discussion which the author had with Elfrieda Fehr at Aachen in 1980. Discussions with Paul Gilmore, Reimond Reiter, and Paul Voda are very much appreciated. This research was supported by Canada Natural Science and Engineering Council grant #A2457.

References

[1] Barendregt, H. The Type Free λ-calculus, Handbook of Math. Logic, J. Barwise, ed., North Holland, 1977.

[2] Church, A. The Calculi of Lambda-conversion. Princeton University Press, 1941.

[3] Ersov, Ju. L. Theorie der Numerierungen I. Zeitschrife für Math. Logik, Bd 19, Heft 4, 1973.

[4] Kanda, A. Fully Effective Solutions of Recursive Domain Equations. Proc. of MFCS'79. Lecture Notes in Computer Science No. 74, 1979.

[5] Kanda, A. Constructive Category Theory I. Proc. of MFCS'81 Lecture Notes in Computer Science No. 118, 1981.

[6] A. Meyer, What Is a Model of λ-calculus? MIT/LCS/TM-201, 1981.

[7] Scott, D. Continuous Lattices, Lecture Notes in Mathematics No. 274, 1971.

[8] Weihrauch and Schäfer, Admissible Representations of Effective CPB's. Proc. of MFCS'81, Lecture Notes in Computer Science, No. 118, 1981.