CLASSES OF NUMERATION MODELS OF λ-CALCULUS

Akira Kanda

Department of Computer Science
The University of British Columbia
Vancouver, B.C. V6T 1W5
Canada

Technical Report 84-4

April 1984

CLASSES OF NUMERATION MODELS OF λ-CALCULUS

Akira Kanda
Department of Computer Science
University of British Columbia
Vancouver, B.C. V6T 1W5
Canada

(April 1984)

Abstract

In [4] the reflexive structures in the category of numeration were studied. It was shown that every numerated reflexive set forms a "numeration model of λ-calculus". In this short note we formalize the concept of numeration models of λ-calculus, and study several interesting subclasses. Even though the class of numeration models does not coincide with the class of numerated reflexive sets, we can show that the class of numeration models with "λ-definability" property is equivalent to the class of numerated reflexive sets with "λ-representability" property. Through this we observe relation between λ-definability and acceptability of numerations discussed in [5].

§1. λ-calculus

The λ-calculus developed by Church [2] is the following formal system: Let V be a countable set of variables.

Definition 1.1. (λ-terms)

1. If $x \in V$ then x is a λ-term
2. If M and L are λ-terms then so is (ML)
3. If $x \in V$ and M is a λ-term then so is (λx.M).

We denote the set of all λ-terms by T.

$\square$

We assume a natural meaning of a λ-term occurring in some other λ-term. 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*. For any terms M, L and a variable x, the result of substituting L for each free occurrence of x in M (and changing bound variables to avoid clashes) is denoted by M[x:=L].

The calculus has the following three reduction rules:

Reduction Rules

$(\alpha)\!:\!(\lambda x.M) \rightarrow (\lambda y.M[x\!:=\!y])$ x is not bound in M and *y does not occur in M*

$(\beta)\!:\!((\lambda x.M)L) \rightarrow M[x\!:=\!L]$

$(\eta)\!:\!(\lambda x.Mx) \rightarrow M$ x does not occur in M.

$\square$

By Godel numbering variables and λ-terms we can realize constructions of λ-terms as a system of recursive functions. Let $v{:}N \to V$ and $\tau{:}N \to T$ be computable bijections. The syntax of λ-terms corresponds to the following system of recursive functions:

$$\textit{is-var}(n) \iff \tau(\mathrm{n}) \in \mathrm{V}$$

$$\textit{is-apply}(n) \iff \tau(\mathrm{n}) = (\mathrm{ML}) \text{ for some } \mathrm{M,L} \in \mathrm{T}$$

$$\textit{is-abst}(n) \iff \tau(\mathrm{n}) = (\lambda \mathrm{x.M}) \text{ for some } \mathrm{x} \in \mathrm{V} \text{ and } \mathrm{M} \in \mathrm{T}.$$

$$\tau(\textit{inc}(\mathrm{n})) = \mathrm{v}(\mathrm{n})$$

$$\textit{is-var}(\mathrm{n}) \Rightarrow \mathrm{v}(\textit{var}(\mathrm{n})) = \tau(\mathrm{n})$$

$$\textit{is-apply}(\mathrm{n}) \Rightarrow \tau(\textit{apply}(\textit{rator}(\mathrm{n}),\textit{rand}(\mathrm{n}))) = \tau(\mathrm{n})$$

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

§2. NUMERATION MODELS OF λ-CALCULUS

Definition 2.1. (Ersov [3]).

A *numeration (of a set* X) is a surjection $\gamma{:}\mathrm{N} \to \mathrm{X}$. A *morphism* from a numeration $\gamma_1{:}N \to X_1$ to another $\gamma_2{:}N \to X_2$ is a function $f{:}X_1 \to X_2$ such that for some recursive function r_f, $f \cdot \gamma_1 = \gamma_2 \cdot r_f$. Such r_f is called a *realization* of f. In case r_f is primitive recursive, we say f is *primitive*.

$\square$

It can readily be seen that numerations and morphisms form a category. (See Ersov [3]).

Let γ:N $\rightarrow$ X be a numeration such that for some numeration $\gamma\uparrow:N \rightarrow Hom(\gamma,\gamma)$, $\gamma \cong \gamma\uparrow$ in the category of numerations. Let υ:N $\rightarrow$ V be the computable bijection discussed in §1. Furthermore let $(\Phi:\gamma\rightarrow\gamma\uparrow, \Psi:\gamma\uparrow\rightarrow\gamma)$ be the isomorphism pair.

An *environment* (or *valuation)* is a primitive morphism from ν to γ. We write Env to denote the set of all environments. Using a Godel numbering $<\psi_i>$ of primitive recursive functions N $\rightarrow$ N, we can introduce a numeration σ:N $\rightarrow$ Env as follows:

$$\sigma_i = \rho \qquad \text{where} \qquad r_\rho = \psi_i.$$

It can readily be seen that updating an environment

$$\rho[x{:=}d](z) = \text{if } x{=}z \text{ then d else } \rho(z)$$

where x$\in$V and d$\in$X has a realization, i.e.

$$\sigma_i[\upsilon(n){:=}\gamma(m)] = \sigma_{update(i,n,m)}$$

for some recursive function update: $N^3 \rightarrow N$. In other word, updating operation is a morphism from $\sigma \times \nu \times \gamma$ to σ.

Definition 2.2.

Let γ be as above. We say γ is a ***numeration model of*** λ***-calculus*** iff the following interpretation function ξ:

$\xi(\tau(n),\sigma_i) :=$ if *is–var*(n) then $\sigma_i(\tau(n))$

else if *is–apply*(n) then

$$\Phi(\xi(\tau(rator(n),\sigma_i))(\xi(\tau(rand(n)),\sigma_i))$$

else if $is\text{-}abst(n)$ then

$$\Psi(\lambda x \in X.\xi(\tau(body(n)),\sigma_i[\tau(bound(n)):=x]))$$

is well-defined and it is a morphism from $\tau \times \sigma$ to γ.

□

It is important to notice that since ξ is a morphism from $\tau \times \sigma$ to γ, $\lambda x \in X.\xi(\tau(body(n)),\ \sigma_i[\tau(bound(n)):=x])$ is a morphism from γ to γ realized by $\lambda m \in N.r_\xi(body(n), update(i, bound(n), m))$. Thus

$$\Psi(\lambda x \in X.\xi(\tau(body(n)),\ \sigma_i[\tau(bound(n)):=x]))$$

is defined. Furthermore the next theorem supports the relevance of this definition:

Theorem 2.3.

Let γ be a numeration model of λ-calculus with an interpretation morphism $\xi{:}\tau\times\sigma\rightarrow\gamma$, then we have:

$$\tau(n) \xrightarrow{\lambda} \tau(m) \textit{ implies } \text{for } \textit{all } i \in N,\ \xi(\tau(n),\sigma_i) = \xi(\tau(m),\sigma_i)$$

where $\tau(n) \xrightarrow{\lambda} \tau(m)$ means that τ(n) can be reduced to τ(m) by one of the reduction rules of λ-calculus.

□

The proof of this theorem is standard and we omit it.

Definition 2.4.

A numeration model γ is λ-*representable* iff there is a recursive function *rep*: N→N such that

$$\gamma(n) = \xi(\tau(rep(n)),\sigma_i) \text{ for } all\ i \in N.$$

A λ-representable numeration model γ is λ-definable iff there is a recursive function *def* such that if a morphism f: $\gamma \rightarrow \gamma$ is realized by a recursive function ϕ_m then

$$f(\gamma(n)) = \xi((\tau(def(m))\tau(rep(n))),\sigma_i) \quad \text{for } all\ i \in N$$

where $<\phi_i>$ is a Godel numbering of partial recursive functions.

$\square$

Note. In a λ-representable numeration model γ: N→X, every element of X can be represented by a closed λ-term. If γ is λ-definable then every morphism $\gamma \rightarrow \gamma$ can be defined by some closed λ-term. Outstanding point here is that we can obtain such λ-term from a Godel number of a recursive function which realizes the morphism.

§3. NUMERATED REFLEXIVE SETS

We outline results of [4]. Main results are as follows: First, a reflexive structure in the category of numeration provides a numeration model of λ-calculus. Second, these structures are exactly constructive extensional combinatory algebras.

Definition 3.1.

Let γ_1: $N \rightarrow X_1$ and γ_2: $N \rightarrow X_2$ be numerations. A numberation γ: $N \rightarrow Hom(\gamma_1, \gamma_2)$ is ***acceptable*** iff there are recursive functions ***realize, numerate***: N→N such that

(1) $r_{\gamma(n)} = \phi_{realize(n)}$

(2) if ϕ_n realizes f: $\gamma_1 \rightarrow \gamma_2$ then $\gamma(numerate(n)) = f$.

□

It can readily be seen that (1) is equivalent to the existence of a (universal) recursive function U: $N^2 \rightarrow N$ such that

$$\gamma(n)(\gamma_1(m)) = \gamma_2(U(n,m)).$$

Also it is known that all acceptable numerations of $Hom(\gamma_1, \gamma_2)$ are recursively isomorphic (see [5]). This means that there is at most one acceptable numeration of $Hom(\gamma_1, \gamma_2)$. Thus we write $(\gamma_1 \rightarrow \gamma_2)$ to denote the acceptable numeration of $Hom(\gamma_1, \gamma_2)$, if any.

Definition 3.2.

A ***numerated reflexive set*** (NRS) is a numeration γ: N→X satisfying:

(1) The acceptable numeration $(\gamma \rightarrow \gamma)$: N→Hom$(\gamma,\gamma)$ exists.

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

It can readily be seen that no non-trivial NRS is finite.

Proposition 3.3.

If γ: N→X is an NRS then it is a numeration model of λ-calculus.

$\square$

The converse of 3.3 does not hold. The existence of an interpretation morphism is not strong enough to prove that $\gamma\dagger$ is acceptable.

We can given an algebraic characterization of NRS's. 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:

$$x_i \in T(X)$$
$$a\in X \Rightarrow a\in T(X)$$
$$A,B\in T(X) \Rightarrow (A\cdot B)\in T(X).$$

We assume that $\cdot$ associates to the left, also we drop $\cdot$ if it does not cause confusion. To denote that a term A has variables $x_0,x_1,\ldots,x_n$, we write $A(x_0,x_1,\ldots,x_n)$. Let ρ: N→T(X) be a Godel numbering of terms.

Definition 3.4.

A ***realizably extensional combinatory algebra*** (RECA) is a 4-tuple $(X,\cdot,\gamma,\rho)$ such that:

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

(2) γ: N→X is a numeration

(3) $\cdot$ is a morphism from $\gamma x\gamma$ to γ.

(4) There is a recursive function λ such that if $\rho(n) = A(x_1,...,x_n)$ then $\gamma(\lambda(\mathrm{n}))$ $= \mathrm{f}$ is a unique element of X satisfying:

$$fy_1...y_n = A(x_1:=y_1,...,x_n:=y_n)$$

where $A(x_1:=y_1,...,x_n:=y_n)$ is the result of substituting y_i for x_i in A $(1 \leq \mathrm{i} \leq \mathrm{n})$.

(5) If for all $\mathrm{d} \in \mathrm{X}$, $d_1 \cdot d = d_2 \cdot d$ then $d_1 = d_2$.

$\square$

Definition 3.5.

An RECA $(X,\cdot,\gamma,\rho)$ is ***computationally complete*** iff there is a recursive function *alg* such that if ϕ_n realizes f: $\gamma \rightarrow \gamma$ then $\sigma(alg(\mathrm{n}))$ is a term with a free variable, say x and

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

$\square$

Proposition 3.6. (Characterization Theorem I)

(1) If $(X,\cdot,\gamma,\rho)$ is a computationally complete RECA then γ is a NRS, where $(\gamma \rightarrow \gamma)$:N$\rightarrow$Hom$(\gamma,\gamma)$ is defined by $(\gamma \rightarrow \gamma)(n) = \Phi(\gamma(n))$ where Φ maps elements of X to functions X$\rightarrow$X defined by $\Phi(x)(y) = x \cdot y$.

(2) If γ: N$\rightarrow$X is a NRS with an isomorphism pair $\Phi:\gamma \rightarrow (\gamma \rightarrow \gamma)$, $\Psi:(\gamma \rightarrow \gamma) \rightarrow \gamma)$ then $(X,\cdot,\gamma,\rho)$ is a computationally complete RECA where $\cdot$ is defined by:

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

$\square$

This proposition is a numeration version of Barendregt's result [1]. It is very important to notice that the class of computationally complete RECA's (or equivalently NRS's) is not the same as the class of numeration models. This indicates a difference between numeration models and set theoretical models. As shown in Barendregt [1], in set theoretical case, models of λ-calculus are the same as extensional combinatory algebras. This difference is due to the following reasons:

(1) $\gamma \cong \gamma\uparrow$ being a numeration model is not strong enough to imply $\gamma\uparrow$: $N \rightarrow Hom(\gamma,\gamma)$ being acceptable.

(2) To obtain the corresponding numerated extensional combinatory algebra from $\gamma\uparrow$, it is crucial to have acceptability of $\gamma\uparrow$.

(3) To obtain a numeration model from a RECA, it is crucial to assume the computational completeness of the RECA.

§4. CHARACTERIZATION OF λ-DEFINABLE NUMERATION MODELS

Even though we can not show good characterization of numeration models of λ-calculus, we can nicely characterize λ-definable models as a sub-class of NRS's.

Definition 4.1.

A NRS γ is ***λ-representable*** iff there is a recursive function ***rep***:N→N such that:

$$\gamma(n) = \xi(\tau(rep(n)),\sigma_i) \quad \text{for } all\ i \in N.$$

where ξ is the interpretation morphism which makes γ a numeration model of λ-calculus.

$\square$

Theorem 4.2.

If a numeration γ is a λ-definable numeration model then it is a λ-representable NRS.

Proof. It is sufficient to show that if γ is a λ-definable numeration model then $\gamma\uparrow$ (remember $(\Phi:\gamma\to\gamma\uparrow,\ \Psi:\gamma\uparrow\to\gamma)$ is an isomorphism pair) is acceptable. We have:

$$\begin{aligned}
&\gamma\uparrow(m)(\gamma(n)) \\
&\quad = \Phi(\gamma(r_\Psi(m)))(\gamma(n)) \\
&\quad = \Phi(\xi(\tau(rep\cdot r_\Psi(m)),\sigma_i))(\xi(\tau(rep(n)),\sigma_i)) \\
&\quad = \xi(\tau(apply(rep\cdot r_\Psi(m),\ rep(n))),\sigma_i) \\
&\quad = \gamma(r_\xi(apply(rep\cdot r_\Psi(m),rep(n)),i)) \\
&\quad = \gamma(U(m,n))
\end{aligned}$$

for some recursive function U: $N^2\to N$. Let ϕ_m realize a morphism f: $\gamma\to\gamma$. We have:

$$\begin{aligned}
f(\gamma(n)) &= \xi((\tau(def(m))\tau(rep(n))),\sigma_i) \\
&= \Phi(\xi(\tau(def(m)),\sigma_i))(\xi(\tau(rep(n)),\sigma_i)) \\
&= \Phi(\gamma(r_\xi(def(m),i)))(\gamma(n)) \\
&= \gamma\uparrow(r_\Phi\cdot r_\xi(def(m),i))(\gamma(n))
\end{aligned}$$

Due to the extensionality

$$\begin{aligned} f &= \gamma\!\uparrow\!(r_\Phi \cdot r_\xi(\mathit{def}(m), i)) \\ &= \gamma\!\uparrow\!(\mathit{numerate}(m)) \end{aligned}$$

for some recursive function *numerate*: N→N. Thus $\gamma\!\uparrow$ is acceptable.

$\square$

Theorem 4.3.

If γ is a λ-representable NRS then it is a λ-definable numeration model of λ-calculus.

Proof. Let $\gamma \cong (\gamma \rightarrow \gamma)$ be a λ-representable NRS. By 3.3 γ is a numeration model of λ-calculus. Let ϕ_m be a recursive function which realizes a morphism f: $\gamma \rightarrow \gamma$. Since γ is a NRS we have:

$$f = (\gamma \rightarrow \gamma)(\mathit{numerate}(m)).$$

Therefore

$$\begin{aligned} \Phi(f) &= \gamma(r_\Phi \cdot \mathit{numerate}(m)) \\ &= \xi(\pi(\mathit{rep} \cdot r_\Phi \cdot \mathit{numerate}(m)), \sigma_i). \end{aligned}$$

Thus

$$\begin{aligned} f(\gamma(n)) &= \Phi(\Psi(f))(\gamma(n)) \\ &= \Phi(\xi(\pi(\mathit{rep} \cdot r_\Psi \cdot \mathit{numerate}(m)), \sigma_i)(\xi(\pi(\mathit{rep}(n)), \sigma_i)) \\ &= \xi((\pi(\mathit{rep} \cdot r_\Psi \cdot \mathit{numerate}(m))\pi(\mathit{rep}(n))), \sigma_i) \\ &= \xi((\pi(\mathit{def}(m))\pi(\mathit{rep}(n))), \sigma_i) \end{aligned}$$

for some recursive function *def*: N→N. Therefore γ is λ-definable.

□

Corollary 4.4. (Characterization Theorem II)

A numeration γ is a λ-definable numeration model iff it is a λ-representable NRS.

□

It is important to observe that we have established the following relationship between acceptability of $\gamma\uparrow$ and λ-definability of a numeration model γ of λ-calculus:

(1) If γ is λ-definable then $\gamma\uparrow$ is acceptable.

(2) If $\gamma\uparrow$ is acceptable and γ is λ-representable then γ is λ-definable.

This correspondance supports the relevance of the concept of acceptable numerations of morphism spaces discussed in [5].

By adding an extra condition to computationally complete RECA, we can characterize λ-definable numeration models. A computationally complete RECA $(X,\cdot,\gamma,\rho)$ is λ-*representable* iff there is a recursive function rep: N→N such that

$$\gamma(n) = \xi(\tau(rep(n)),\sigma_i) \quad \text{for } all\ i \in N.$$

Theorem 4.5. (Characterization Theorem III)

A numeration γ: N→X is a λ-definable numeration model iff (the corresponding) $(X,\cdot,\gamma,\rho)$ is a λ-representable computationally complete RECA.

□

Unfortunately, there is no known numeration model which is λ-definable. The language of λ-calculus is too weak to represent all computable elements of D_∞ of Scott [6].

One possible solution to this problem is to enrich the language by introducing finite constant symbols. Then by adding interpretation of these constant symbols to ξ we can define the concept of numeration models of this enriched λ-calculus. It is easy to observe that all results of §2, §3, and §4 hold for the enriched λ-calculus. Furthermore it can readily be seen that a numeration model due to the computable elements of D_∞ (as discussed in [4]) is λ-definable for a suitably enriched λ-calculus (see [6] for details).

Of course by introducing countably infinite constant symbols and providing a constant symbol for each element of a NRS χ: N$\rightarrow$X, we can make χ λ-representable for the expanded λ-calculus. But this contradicts to the fundamental role of λ-calculus, which is to define functions via application and abstraction and to reason about these functions.

ADDENDUM

It is known that by Gödel numbering the term model of Church-Barendregt using Gödel numbering τ of λ-terms, we can obtain a numeration model of λ-calculus (see [4] or Visser [7]). But this model is not λ-definable because an equivalence class $[t]$ with t being an open term can not be represented by a closed term, in general.

ACKNOWLEDGEMENT

This research was motivated by very simulating discussions which the author had with Longo and Moggi at Pisa in 1984. Most thanks are due to them. This research was supported by Canada Natural Science and Engineering Research 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. Zeitshrife fur Math. Logik, Bd 19, Heft 4, 1973.

[4] Kanda, A., Numeration Models of λ-calculus*. Tech. Rep. 83-3, University of British Columbia, 1983. (To appear in ZML).

[5] Kanda, A., Acceptable Numerations of Function Spaces. Tech. Rep. 83-12, Univeresity of British Columbia, 1983. (To appear in ZML)

[6] Scott, D., Data Types as Lattices, Lecture Note, Amsterdam, 1974.

*An extended abstract of this paper is published in the Proc. of the IXth Coloquium on Trees in Algebra and Programming. Cambridge University Press, 1984.