ソフトウェア基礎論　30−1
(1989. 9. 8)

パラメトリック・ポリモルフィズム における再帰的型定義

Recursive Type Definition in Parametric Polymorphism

長谷川立
Ryu HASEGAWA
京都大学数理解析研究所
RIMS, Kyoto University

0. Introduction

The second order polymorphic lambda calculus (we say hereafter simply polymorphism) is an extension of the simple typed lambda calculus. We can use type variables in deduction of terms, handling them through type abstraction $\Lambda t.M$ and its elimination $M[\sigma]$. A term of the form $\Lambda t.M$ can be considered as a type-indexed families of terms, which are defined for all types.

Here is a philosophical standpoint. As stated in [6], Strachey distinguished parametric polymorphism from ad hoc polymorphism. A parametric function must have a uniform definition for each type, while an ad hoc function may be only a bundle of functions, which has various definitions according to their types. In polymorphism, we aim at taking up only parametric functions.

In semantics, what should we do to keep ad hoc functions out? Pursuing a set-theoretical model, Reynolds provided a presumption in [6]. In his model, the meaning of a function type $[\![\Gamma \vdash \sigma \Rightarrow \tau]\!]\eta$ is the set of all functions from $[\![\Gamma \vdash \sigma]\!]\eta$ to $[\![\Gamma \vdash \tau]\!]\eta$, and the meaning of a universal type $[\![\Gamma \vdash \forall t.\sigma]\!]\eta$ is a set of type-indexed families of functions. If it is, however, the whole collection of families, it may be too large too be a set. Therefore we must restrict it to smaller one, and he thought that if collecting only parametric functions we may evade cardinality explosion. Unfortunately his attempt failed, as proved by himself in [7], and it stimulates the development of various notions of models for polymorphism [2], [5], [7]. In this paper we adopt the so-called Bruce-Meyer models, and develop Reynolds' semantical notion of parametricity in them. Our main assertion is that it is not only mere philosophy, but has some concrete applications.

In this paper, we concentrate on the problem of recursive type definitions. It is well known that many useful types can be defined as initial fixed points of domain equations, like natural numbers, lists, and trees, and they are especially explored well in the category of complete partial orders [4]. For instance, in a category with a terminal object and coproducts, the type of natural numbers is an initial fixed point of the equation $X \approx 1+X$.

After some preliminaries, in Theorem 3.4., we prove that a category constructed from a model has initial fixed points if and only if a certain kind of parametric condition holds in the model.

1. Syntax, semantics, and parametricity.

1.1. Syntax.

The second order order polymorphic lambda calculus (or polymorphism) is defined by the following data.

A type has the form $\Gamma \vdash \sigma$, where Γ is a finite set of type variables. The set of all types is the least closure of the following rules,

(Ty-1) $\Gamma \vdash t \qquad (t \in \Gamma)$

(Ty-2) $\dfrac{\Gamma \vdash \sigma \quad \Gamma \vdash \tau}{\Gamma \vdash \sigma \Rightarrow \tau}$

(Ty-3) $\dfrac{\Gamma, t \vdash \sigma}{\Gamma \vdash \forall t.\sigma}$

Sometimes '$\Gamma \vdash$' will be omitted.

Syntactical type assignments to ordinary variables have the form

$x_1: \Gamma \vdash \sigma_1, \ \cdots \ , x_n: \Gamma \vdash \sigma_n$

where all two x_i and x_j are disjoint. Note that all $\Gamma \vdash$ is common, and although it is written as a sequence, it is taken for a set.

A term has the form

$\Theta \vdash M: \Gamma \vdash \sigma$

where Θ is syntactical type assignments in which all ordinary variables are typed to those which begin with '$\Gamma \vdash$'. The set of all terms is the least closure of the following rules,

(Te-1) $\Theta \vdash x: \Gamma \vdash \sigma \quad (x: \Gamma \vdash \sigma \in \Theta)$

(Te-2) $\dfrac{\Theta,\ x: \Gamma \vdash \sigma \vdash M: \Gamma \vdash \tau}{\Theta \vdash \lambda x^{\sigma}.\ M : \Gamma \vdash \sigma \Rightarrow \tau}$

(Te-3) $\dfrac{\Theta \vdash M: \Gamma \vdash \sigma \Rightarrow \tau \quad \Theta \vdash N: \Gamma \vdash \sigma}{\Theta \vdash MN: \Gamma \vdash \tau}$

(Te-4) $\dfrac{\Theta^{+t} \vdash M: \Gamma, t \vdash \sigma}{\Theta \vdash \Lambda t.M: \Gamma \vdash \forall t.\sigma}$

where for syntactical type assignments $\Theta = \underline{x: \Gamma \vdash \sigma}$ (we use an underline to denote a finite set), in which Γ does not include t, $\Theta^{+t} = \underline{x: \Gamma, t \vdash \sigma}$

(Te-5) $\dfrac{\Theta \vdash M: \Gamma \vdash \forall t.\sigma}{\Theta \vdash M[\tau]: \Gamma \vdash \sigma[t:=\tau]}$

where $\Gamma \vdash \tau$ is a type, and $\sigma[t:=\tau]$ is one in which we substituted τ for t in σ. We will make use of some abbreviations of $\Theta \vdash M: \Gamma \vdash \sigma$ into , e.g. in the shortest case, M.

We can equate two terms $\Theta \vdash M : \Gamma \vdash \sigma$ and $\Theta' \vdash N : \Gamma' \vdash \sigma'$ only if $\Theta = \Theta'$ and $\Gamma \vdash \sigma = \Gamma' \vdash \sigma'$. Equality between two terms are the least congruence relation including the next four conversions,

(β) $(\lambda x.M)N = M[x:=N]$

(η) $\lambda x.Mx = M$

where x does not occur freely in M.

(Type β) $(\Lambda t.M)[\tau] = M[t:=\tau]$

(Type η) $\Lambda t.M[t] = M$

where t does not occur freely in M.

1.2. Semantics.

T is a quadruple $<\mathfrak{T}, [\mathfrak{T}=>\mathfrak{T}], =>, \forall>$ where $\mathfrak{T}$ is a non-empty set, $[\mathfrak{T}=>\mathfrak{T}]$ is a set of functions on $\mathfrak{T}$ to $\mathfrak{T}$, $=>: \mathfrak{T}^2 -> \mathfrak{T}$, and $\forall : [\mathfrak{T}=>\mathfrak{T}] -> \mathfrak{T}$. For $F \in [\mathfrak{T}=>\mathfrak{T}]$, we write $\forall X.F(X)$ for $\forall(F)$.

For each $A \in \mathfrak{T}$, a possibly empty set D_A is provided.

For each $A, B \in \mathfrak{T}$,

$$\begin{tikzcd} D_{A \Rightarrow B} \arrow[r, shift left, "\Phi^{\Rightarrow}_{A,B}"] & {[D_A => D_B]} \arrow[l, shift left, "\Psi^{\Rightarrow}_{A,B}"] \end{tikzcd}, \qquad \Phi^{\Rightarrow} \circ \Psi^{\Rightarrow} = \mathrm{id},$$

where $[D_A=>D_B]$ is a subset of functions from D_A to D_B. For each $F \in [\mathfrak{T}=>\mathfrak{T}]$,

$$\begin{tikzcd} D_{\forall X.F(X)} \arrow[r, shift left, "\Phi^{\forall}_F"] & {[\Pi X^{\mathfrak{T}}.D_{F(X)}]} \arrow[l, shift left, "\Psi^{\forall}_F"] \end{tikzcd}, \qquad \Phi^{\forall} \circ \Psi^{\forall} = \mathrm{id},$$

where $[\Pi X^{\mathfrak{T}}.D_{F(X)}]$ is a subset of $\Pi X^{\mathfrak{T}}.D_{F(X)}$.

We call a triple $\xi = <T, D, <\Phi, \Psi>>$ a <u>second order functional domain</u>. If in addition $\Psi \circ \Phi = \mathrm{id}$, we say that the domain is <u>extensional</u>. From now on, all considered domain is extensional.

A type $t_1, \ ... \ , t_n \vdash \sigma$ is interpreted in $\mathfrak{T}^n --> \mathfrak{T}$, as :

(Ty-1) $[\![\Gamma \vdash t]\!](\Gamma := \underline{A}) = A$,

(Ty-2) $[\![\Gamma \vdash \sigma => \tau]\!]\eta = [\![\Gamma \vdash \sigma]\!]\eta => [\![\Gamma \vdash \tau]\!]\eta$,

(Ty-3) $[\![\Gamma \vdash \forall t.\sigma]\!]\eta = \forall X.[\![\Gamma, t \vdash \sigma]\!]\eta(t:=X)$.

Here we use η for a type environment.

A term $\Theta \vdash M : \Gamma \vdash \tau$ where $\Gamma = t_1, \ ... \ , t_n$, and $\Theta = x_1 : \Gamma \vdash \sigma_1, \ ... \ , x_n : \Gamma \vdash \sigma_n$, is interpreted in

$$\Pi X_1^{\mathfrak{T}} ... \Pi X_n^{\mathfrak{T}}(\underline{[\![\Gamma \vdash \sigma]\!](\Gamma := X)} --> [\![\Gamma \vdash \tau]\!](\Gamma := \underline{X}))$$

where $\underline{[\![\Gamma \vdash \sigma]\!](\Gamma := X)} = [\![\Gamma \vdash \sigma_1]\!](\Gamma := \underline{X}) * ... * [\![\Gamma \vdash \sigma_n]\!](\Gamma := \underline{X})$, as :

(Te-1) $[\![\Theta \vdash x : \Gamma \vdash \sigma]\!]\eta(\underline{x:=a}) = a$,

(Te-2) $[\![\Theta \vdash \lambda x.^{\sigma} M : \Gamma \vdash \sigma => \tau]\!]\eta\rho = \Psi^{\Rightarrow}(\lambda d.^{[\![\sigma]\!]\eta}[\![M]\!]\eta\rho(y:=d))$,

(Te-3) $[\![\Theta \vdash MN : \Gamma \vdash \tau]\!]\eta\rho = \Phi^{\Rightarrow}([\![M]\!]\eta\rho)([\![N]\!]\eta\rho)$,

$$\text{(Te-4)} \quad \llbracket \Theta \vdash \Lambda t.M : \Gamma \vdash \forall t.\sigma \rrbracket \eta\rho = \Psi^{\forall}(\lambda X.^{\mathfrak{T}} \llbracket M \rrbracket \eta(t:=X)\rho),$$

$$\text{(Te-5)} \quad \llbracket \Theta \vdash M[\tau] : \Gamma \vdash \sigma[t:=\tau] \rrbracket \eta\rho = \bar{\Phi}^{\forall}(\llbracket M \rrbracket \eta\rho)(\llbracket \tau \rrbracket \eta).$$

Here we use ρ for a term environment.

If the interpretation above is possible, the extensional domain is called a second order model or simply a model. The reader can refer to [1], [2] for the detail.

1.3. Parametricity.

For A, B $\in \mathfrak{T}$, a relation χ between A and B is a subset of $D_A * D_B$ and denoted by $A \xrightarrow{\chi} B$. For $a \in D_A$ and $b \in D_B$, we write $a \xrightarrow{\chi} b$ for $(a,b) \in \chi$.

Let $\mathfrak{R}$ be the collection of all relations, and $\mathfrak{R}_0$ a subset of $\mathfrak{R}$. We assume that $F \in [\mathfrak{T} => \mathfrak{T}]$ can be extended to a function on $\mathfrak{R}_0$ to $\mathfrak{R}$, such that

$$A \xrightarrow{\chi} B \in \mathfrak{R}_0 \quad => \quad F(A) \xrightarrow{F(\chi)} F(B).$$

Definition 1.1.

Set $\mathfrak{R}$, $\mathfrak{R}_0$, and F as above.

i) $D_{\forall X.F(X)}$ is parametric w.r.t. $\mathfrak{R}_0$ if and only if for all $a \in D_{\forall X.F(X)}$ and $A \xrightarrow{\chi} B \in \mathfrak{R}_0$,

$$\bar{\Phi}^{\Rightarrow}(a)(A) \xrightarrow{F(\chi)} \bar{\Phi}^{\forall}(a)(B).$$

ii) A model ξ is parametric w.r.t. $\mathfrak{R}_0$ if and only if for all $F \in [\mathfrak{T} => \mathfrak{T}]$, F can be extended to a function of $\mathfrak{R}_0 \dashrightarrow \mathfrak{R}$, and $D_{\forall X.F(X)}$ is parametric w.r.t. $\mathfrak{R}_0$.

This definition is similar to that of Reynolds, [6].

2. Categorical establishment.

2.1. Categorification.

From a model ξ, we construct a small category C_ξ. The idea is simple. $\mathrm{Obj}(C_\xi) = \mathfrak{T}$ and $\mathrm{Hom}_{C_\xi}(A,B) = D_{A \Rightarrow B}$. An identity arrow $1_A : A \dashrightarrow A$ is defined as

$$1_A = \llbracket \lambda x.^{t_A} x \rrbracket \eta\rho \tag{1}$$

and a composition $g \circ f : A \dashrightarrow C$ for $f : A \dashrightarrow B$ and $g : B \dashrightarrow C$ is

$$g \circ f = \llbracket \lambda x.^{t_A} x_g(x_f x) \rrbracket \eta\rho \tag{2}$$

These definitions show the main technique of proofs in this paper. In (1) one prepares a new type variable t_A and deduces a term

$$\vdash \lambda x.^{t_A} x : t_A \vdash t_A => t_A.$$

And its interpretation under the environment $\eta = (t_A := A)$ and $\rho = \phi$, is defined to be 1_A. In (2) three type variables t_A, t_B, and t_C, and two ordinary variables x_f, x_g, are prepared, and a term

$$x_f : \Gamma \vdash t_A => t_B,\ x_g : \Gamma \vdash t_B => t_C \vdash \lambda x.^{t_A} x_g(x_f x) : \Gamma \vdash t_A => t_C,$$

where $\Gamma = t_A, t_B, t_C$, is interpreted under the environment $\eta = (t_A := A)(t_B := B)(t_C := c)$ and $\rho = (x_f := f)(x_g := g)$. Then it is to be the composition $g \circ f$.

Henceforth we will not mention to the content of each environment, anticipating the reader could recover the ambiguity from the suffixes, and even may use the same η or ρ for different environments.

Using this technique, e.g. in Definition 1.1., one may write

$$[\![x_a[t_A]]\!] \eta\rho \xrightarrow{F(X)} [\![x_a[t_B]]\!] \eta\rho$$

for $\bar{\Phi}^{\forall}(a)(A) \xrightarrow{F(X)} \bar{\Phi}^{\forall}(a)(B)$.

Proposition 2.1.

C_ξ is a small category.

2.2. Representable functors

Take a type $\Gamma, t \vdash \sigma$. In a type environment, fixing the assignments for Γ and not determining one for t, we obtain a function $F \in [\mathcal{T}=>\mathcal{T}]$, as

$$F = [\![\Gamma, t \vdash \sigma]\!] \eta(t:=-) \text{ or simply } [\![\Gamma, t \vdash \sigma]\!] \eta$$

where η is a fixed environment for Γ. We will extend it to an endofunctor on C_ξ.

Definition 2.2.

Let F be $[\![\Gamma, t \vdash \sigma_F]\!] \eta$. If t appears only positively (negatively) in σ_F, F can be extended to a covariant endofunctor on C (or contravariant resp.).

We must extend F to be an arrow function. First for any term $M : \Gamma \vdash \rho => \rho'$, inductively define a term

$$\sigma_F[t:=M] : \Gamma \vdash \sigma_F[t:=\rho] => \quad \sigma_F[t:=\rho']$$

(in the covariant case). And for an arrow $f : A \dashrightarrow B$, put

$$F(f) = [\![\sigma_F[t:=x_f]]\!] \eta\rho.$$

i) $\sigma_F = s \ (s \in \Gamma)$,

$$\sigma_F[t:=M] = \lambda x.^{s}\ x$$

ii) $\sigma_F = t$,

$$\sigma_F[t:=M] = M$$

iii) $\sigma_F = \sigma_G => \sigma_H$ where σ_G is contravariant and σ_H is covariant,

$$\sigma_F[t:=M] = \lambda x.^{\tau} \lambda y.^{\tau'}\ \sigma_H[t:=M](x(\sigma_G[t:=M](y))),$$

where $\tau = \sigma_F[t:=\rho]$, and $\tau' = \sigma_G[t:=\rho']$.

iv) $\sigma_F = \forall u.\sigma_G$,

$$\sigma_F[t:=M] = \lambda x.^{\tau} \Lambda u.(\sigma_G[t:=M](x[u])),$$

where $\tau = \sigma_F[t:=\rho]$.

Similar for the contravariant case.

We call such F a representable (covariant or contravariant) functor.

2.3. Representable relation functions.

Take a type $\Gamma, t \vdash \sigma$ again, in which case t may occur both positively and negatively in σ. This time we extend it to a function from $\mathfrak{R}$ to $\mathfrak{R}$. We will call such F a <u>representable relation function</u>.

<u>Definition 2.3.</u>

$F = [\![\Gamma, t \vdash \sigma_F]\!]\eta$ can be extended to a function of $\mathfrak{R} \dashrightarrow \mathfrak{R}$, such that for any $A \xrightarrow{\chi} B \in \mathfrak{R}$, $F(A) \xrightarrow{F(\chi)} F(B)$.

The definition is by the induction on the structure of σ_F.

i) $\sigma_F = s$ $(s \in \Gamma)$,

$F(\chi) = [\![s]\!]\eta(t:=\chi) = 1_{[\![s]\!]\eta}$ (identity relation).

ii) $\sigma_F = t$,

$F(\chi) = [\![t]\!]\eta(t:=\chi) = \chi$.

iii) $\sigma_F = \sigma_G \Rightarrow \sigma_H$,

$F(\chi) = G(\chi) \Rightarrow H(\chi)$

where for $f \in F(A)$ and $g \in F(B)$, $f \xrightarrow{G(\chi) \Rightarrow H(\chi)} g$ iff for any $a \xrightarrow{G(\chi)} b$, $\vec{\Phi}(f)(a) \xrightarrow{H(\chi)} \vec{\Phi}(g)(b)$ or equivalently $[\![x_f x_a]\!]\eta\rho \xrightarrow{H(\chi)} [\![x_g x_b]\!]\eta\rho$.

iv) $\sigma_F = \forall u.\sigma_G$,

for any $f \in F(A)$ and $g \in F(B)$, $f \xrightarrow{F(\chi)} g$ iff for any $C \in \mathfrak{T}$,

$\Phi^{\forall}(f)(C) \xrightarrow{G_C(\chi)} \Phi^{\forall}(g)(C)$

where $G_C(\chi) = [\![\Gamma, u, t \vdash \sigma_G]\!]\eta(u:=C)(t:=\chi)$, or equivalently $[\![x_f[t_C]]\!]\eta\rho \xrightarrow{G_C(\chi)} [\![x_g[t_C]]\!]\eta\rho$.

We are especially interested in a set of relations called $\mathfrak{R}_{Ar}$.

<u>Definition 2.4.</u> $\mathfrak{R}_{Ar}$ consists of all relations of the form $A \xrightarrow{|f|} B$ where $f : A \dashrightarrow B$ is an arrow in C_ξ, and for $a \in A$ and $b \in B$,

$a \xrightarrow{|f|} b$ iff $\vec{\Phi}(f)(a) = b$ (or $[\![x_f x_a]\!]\eta\rho = [\![x_b]\!]\eta\rho$).

<u>Proposition 2.5.</u>

Let F be a representable covariant functor (hence a representable relation function). Then for any $|f| \in \mathfrak{R}_{Ar}$,

$F(|f|) = |F(f)|$

Another useful tool for relating a functor to a relation function is : for G and H representable covariant functors,

$g \xrightarrow{G(|f|) \Rightarrow H(|f|)} h$ where $A \xrightarrow{|f|} B$

if and only if

$$\begin{tikzcd}
G(A) \arrow[r, dashed, "g"] \arrow[d, "G(f)"'] \arrow[dr, phantom, "\circlearrowleft"] & H(A) \arrow[d, "H(f)"] \\
G(B) \arrow[r, dashed, "h"] & H(B)
\end{tikzcd}$$

3. Initial fixed point .

3.1. Fixed point.

Definition 3.1.

Let C be a category and F an endofunctor on C.

i) An object M of C is called a pre-fixed point of F, if there exists an arrow u : F(M) --> M.

ii) M is called a fixed point of F, if in addition there exists an arrow u': M --> F(M), and $u' \circ u = id_{F(M)}$ and $u \circ u' = id_M$.

We are interested in the case that F is representable, and in a special fixed point called initial fixed point. To explain it we need the category F-Alg defined below.

Definition 3.2.

Let F be an endofunctor on a category C. The category of F-algebras F-Alg is given by the following data :

i) an object is (A, f) such that f : F(A) --> A in C.

ii) an arrow g : (A, f) --> (A', f') is an arrow g : A --> A' in C such that

$$\begin{tikzcd}
F(A) \arrow[r, dashed, "F(g)"'] \arrow[d, "f"] & F(A') \arrow[d, "f'"] \\
A \arrow[r, dashed, "g"] & A'
\end{tikzcd}$$

commutes.

Proposition 3.3.

If F-Alg has an initial object (M, u), M is a fixed point of F.

For the proof of Proposition 3.3. the reader can refer to [4], [8], [10]. We call that fixed point an initial fixed point of F. The importance of initial fixed points is studied in [4]. If (M, u) is just only a pre-initial fixed point of F-Alg, which is defined by weakening 'unique' to 'at least one' in the universal condition of initiality, we say M a pre-initial (pre-)fixed point.

3.2. Initial fixed point in parametric polymorphism

Here is our main theorem.

Theorem 3.4.

Let F be a representable endofunctor on C_{ξ}.

i) $M = \forall X.((F(X) \Rightarrow X) \Rightarrow X)$ is a pre-initial fixed point of F.

ii) M is an initial fixed point of F, if and only if M is parametric w.r.t. $\mathcal{R}_{Ar}$.

(Proof)

i) Let F be represented by $\Gamma, t \vdash \sigma_F$, and put $\sigma_M = \forall t.((\sigma_F \Rightarrow t) \Rightarrow t)$. For each f : F(A) --> A, we define an arrow p_f: M --> A as

$$p_f = [\![\lambda x.^{\sigma_M} x[t_A] x_f]\!] \eta\rho,$$

and define u : F(M) -->M as

$$u = [\![\lambda x.^{\sigma_F[t:=\sigma_M]} \Lambda t. \lambda y.^{\sigma_F \Rightarrow t} y(\sigma_F[t:=M_{p_f}](x)) \qquad (3)$$

where $M_{p_f} = \lambda x.^{\sigma_M} x[t] y$.

Then by direct calculation,

$$\begin{tikzcd} F(M) \arrow[r, "F(p_f)"] \arrow[d, "u"] & F(A) \arrow[d, "f"] \\ M \arrow[r, "p_f"] & A \end{tikzcd}$$

commutes for any f : F(A) --> A. Therefore M is a pre-initial fixed point of F.

ii) (=>)

Take any $n \in D_M$. We must show that

$$[\![x_n[t_A]]\!] \eta\rho \xrightarrow{(F(|f|) \Rightarrow |f|) \Rightarrow |f|} [\![x_n[t_B]]\!] \eta\rho \qquad (4)$$

for any $A \xrightarrow{|f|} B$.

Take any $g \xrightarrow{F(|f|) \Rightarrow |f|} h$. Using the remark just below Proposition2.5., the inner, right trapezoid in the following diagram commutes.

$$\begin{tikzcd} & & F(B) \arrow[dddd, "h"] \\ F(M) \arrow[urr, "F(p_h)"] \arrow[r, "F(p_g)"'] \arrow[dd, "u"] & F(A) \arrow[ur, "F(f)"'] \arrow[dd, "g"] & \\ & & \\ M \arrow[r, "p_g"] \arrow[drr, "p_h"'] & A \arrow[dr, "f"] & \\ & & B. \end{tikzcd}$$

Then, since M is supposed to be initial in F-Alg, all displayed diagrams in this figure commute. In particular, the lower triangle shows that for any $n \in D_M$,

$$\Phi^{\Rightarrow}(p_g)(n) \xrightarrow{|f|} \Phi^{\Rightarrow}(p_h)(n),$$

that is,

$$[\![x_n[t_A]x_g]\!]\eta\rho \xrightarrow{|f|} [\![x_n[t_B]x_h]\!]\eta\rho.$$

thus (4) is proved.

(<=)

First prove that p_u: M --> M is id_M. Take any g : F(A) --> A, and

$$u \xrightarrow{F(|p_g|) \Rightarrow |p_g|} g.$$

Since M is parametric w.r.t. $\mathcal{R}_{Ar}$, for any $n \in D_M$,

$$[\![x_n[\sigma_M]M_u]\!]\eta\rho \xrightarrow{|p_g|} [\![x_n[t_A]x_g]\!]\eta\rho$$

where M_u is the lambda term defining u in (3). Thus

$$[\![(x_n[\sigma_M]M_u)[t_A]x_g]\!]\eta\rho = [\![x_n[t_A]x_g]\!]\eta\rho$$

Since A, g, and n are arbitrary, taking λx_g, Λt , and λx_n in this order,

$$p_u = \mathrm{id}_M. \tag{5}$$

Next prove that for any h : (M, u) --> (A, g) in F-Alg, $p_g = h \circ p_u$. From the definition of h

$$u \xrightarrow{F(|h|) \Rightarrow |h|} g.$$

Then similar to the above, for any $n \in D_M$,

$$[\![x_n[\sigma_M]M_u]\!]\eta\rho \xrightarrow{|h|} [\![x_n[t_A]x_g]\!]\eta\rho,$$

that is,

$$[\![x_h(x_n[\sigma_M]\ M_u)]\!]\eta\rho = [\![x_n[t_A]x_g]\!]\eta\rho.$$

Taking λx_n,

$$h \circ p_u = p_g. \tag{6}$$

From (5) and (6), we obtain $h = p_g$, which shows that M is an initial fixed point.

Example 3.5.

Let $\mathfrak{z}$ be a model parametric w.r.t. $\mathcal{R}_{Ar}$. Then $C_{\mathfrak{z}}$ has a terminal object 1 and coproducts (it will be showed in the forthcoming paper). So

$$F(X) = 1+X$$

is a representable functor, and from Theorem 3.5., N = ∀X.(((1+X)=>X)=>X) is an initial fixed point of F. N is a natural numbers object in the sense of [3], and equivalent to the usual encoding ∀X.((X=>X)=>X=>X).

References

[1] Val Breazu-Tannen and Thierry Coquand. Extensional models for polymorphism, in H. Ehrig et al. eds., TAPSOFT '87, LNCS 250(1987), Springer-Verlag, pp.291--307.

[2] Kim B. Bruce and Albert R. Meyer. The semantics of second order lambda calculus, in G. Kahn, D. B. MacQueen, and G. Plotkin eds., Semantics of Data Types, LNCS 173(1984), Springer-Verlag, pp.131--144.

[3] J. Lambek and P. J. Scott. Introduction to higher order categorical logic, 1986, Cambridge Univ. Press.

[4] Daniel J. Lehman and Michael B. Smyth. Algebraic specification of data types : a synthetic approach, Mathematical Systems Theory 14(1981), 97--139.

[5] José Meseguer. Relating models of polymorphism, Report No. CSLI-88-133, 1988, Center for the Study of Language and Information, SRI International.

[6] John C. Reynolds. Types, abstraction, and parametric polymorphism, in R. E. A. Mason ed., Information Processing '83, pp.513--523.

[7] John C. Reynolds. Polymorphism is not set-theoretic, in G. Kahn, D. B. MacQueen, and G. Plotkin eds., Semantics of Data Types, LNCS 173(1984), Springer-Verlag, pp.145--156.

[8] J. C. Reynolds and G. D. Plotkin. On functors expressible in the polymorphic typed lambda calculus, ECS-LFCS-88-53, 1988, LFCS, Department of Computer Science, University of Edinburgh.

[9] R. A. G. Seely. Categorical semantics for higher order polymorphic lambda calculus, Journal of Symbolic Logic 52(1987), 969--989.

[10] M. B. Smyth and G. D. Plotkin. The category theoretic solution of recursive domain equations, SIAM Journal of Computing 11(1982), 761--783.