This is a preprint of a paper to be given at the Normalization
by Evaluation Workshop (NBE'98) in Aarhus, Denmark.

Normalization and Functor Categories*

John C. Reynolds
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3890, USA
john.reynolds@cs.cmu.edu

May 9, 1998

Abstract

There is a functor-category model of the simply typed lambda calculus such that the long β-η-normal form of any expression can be obtained from its meaning.

Let Θ be the set $\Theta ::= \iota \mid \Theta \to \Theta$ of types of the simply typed lambda calculus (with a single primitive type ι). Let $\mathcal{V}$ be the set of variables. Then we define Σ to be the category of type assignments and type-preserving variable maps:

$$\mathrm{Ob}\,\Sigma = \{\, \pi \mid \pi \in F \to \Theta \text{ for some } F \overset{\text{fin}}{\subseteq} \mathcal{V} \,\}$$
$$\pi \xrightarrow[\Sigma]{} \pi' = \{\, \phi \mid \phi \in \mathrm{dom}\,\pi \to \mathrm{dom}\,\pi' \text{ and } \pi = \phi\,;\pi' \,\},$$

with composition and identities inherited from SET.

It is well known that the functor category SET^{Σ} is Cartesian closed (actually, for any category Σ). The exponentiation of two functors F, G from Σ to SET is the functor $F \Rightarrow G$ whose action on objects of Σ is

$$(F \Rightarrow G)\pi = \mathrm{Hom}_{\Sigma}\,\pi \times F \xrightarrow[\mathrm{SET}^{\Sigma}]{} G,$$

where $\mathrm{Hom}_{\Sigma} \in \Sigma^{\mathrm{op}} \to (\Sigma \to \mathrm{SET})$ is the Curried hom-functor for the category Σ. Thus, if $\eta \in (F \Rightarrow G)\pi$ then $\eta\pi' \in (\pi \xrightarrow[\Sigma]{} \pi') \times F\pi' \to G\pi'$. The action of $F \Rightarrow G$ on morphisms of Σ is

$$(F \Rightarrow G)(\phi \in \pi \xrightarrow[\Sigma]{} \pi')\big(\eta \in (F \Rightarrow G)\pi\big)\pi''\langle \phi' \in \pi' \xrightarrow[\Sigma]{} \pi'', a \in F\pi''\rangle = \eta\pi''\langle \phi\,;\phi', a\rangle.$$

*This research was sponsored in part by National Science Foundation Grant CCR-9804014.

The fact that SET^{Σ} is Cartesian closed leads directly to a semantics for the simply typed lambda calculus. Types denote functors from Σ to SET, and the type constructor $\rightarrow$ is interpreted by $\Rightarrow$:

$$[\![\theta \rightarrow \theta']\!] = [\![\theta]\!] \Rightarrow [\![\theta']\!].$$

Type assignments denote functors defined by products in SET^{Σ}

$$[\![\pi]\!] = \prod_{v \in \mathrm{dom}\,\pi}^{\mathrm{SET}^{\Sigma}} [\![\pi v]\!].$$

Since these products are defined pointwise, we have

$$[\![\pi]\!]\pi'(v \in \mathrm{dom}\,\pi) = [\![\pi v]\!]\pi'$$
$$[\![\pi]\!](\phi \in \pi' \xrightarrow[\Sigma]{} \pi'')(\eta \in [\![\pi]\!]\pi')(v \in \mathrm{dom}\,\pi) = [\![\pi v]\!]\phi(\eta v).$$

When an expression e satisfies the typing judgement $\pi \vdash e : \theta$, its meaning $[\![e]\!]_{\pi\theta}$ is a natural transformation from $[\![\pi]\!]$ to $[\![\theta]\!]$. These natural transformations satisfy the following semantic equations:

$$[\![v]\!]_{\pi,\pi v}\pi'\eta = \eta v$$
$$[\![e\,e']\!]_{\pi\theta'}\pi'\eta = [\![e]\!]_{\pi,\theta\rightarrow\theta'}\pi'\eta\pi'\langle I^{\Sigma}_{\pi'}, [\![e']\!]_{\pi\theta}\pi'\eta\rangle$$
$$[\![\lambda v.\ e]\!]_{\pi,\theta\rightarrow\theta'}\pi'\eta\pi''\langle\phi \in \pi' \xrightarrow[\Sigma]{} \pi'', a \in [\![\theta]\!]\pi''\rangle = [\![e]\!]_{[\pi|v:\theta],\theta'}\pi''[\,[\![\pi]\!]\phi\eta \mid v\colon a\,].$$

Here $I^{\Sigma}_{\pi'}$ is the identity morphism on π' in Σ, and $[\,f \mid v\colon a\,]$ denotes the function with domain $\mathrm{dom}\,f \cup \{v\}$ that maps v into a and all other arguments w into fw.

To complete our semantics, we need only define the functor $[\![\iota]\!]$ that gives meaning to the primitive type ι. Before doing so, however, we must digress to define some sets of expressions in the simply typed lambda calculus.

To avoid confusing expressions of the lambda calculus with metalinguistic abstractions and applications, we write appl for the function that maps expressions e and e' into the application $e\,e'$, and abstr for the function that maps a variable v and an expression e into the abstraction $\lambda v.\ e$. Then we define $\mathcal{M}$ and $\mathcal{N}$ to be the least families of sets of expressions, each indexed by a type assignment π and a type θ, such that

$$\mathcal{M}_{\pi\theta} = \{\, v \mid v \in \mathrm{dom}\,\pi \text{ and } \pi v = \theta \,\} \cup \{\, \mathsf{appl}(e, e') \mid e \in \mathcal{M}_{\pi,\hat{\theta}\rightarrow\theta} \text{ and } e' \in \mathcal{N}_{\pi\hat{\theta}} \,\}$$
$$\mathcal{N}_{\pi\iota} = \mathcal{M}_{\pi\iota}$$
$$\mathcal{N}_{\pi,\theta\rightarrow\theta'} = \{\, \mathsf{abstr}(v, e) \mid v = \mathsf{new}(\mathrm{dom}\,\pi) \text{ and } e \in \mathcal{N}_{[\pi|v:\theta],\theta'} \,\}.$$

Here $\mathsf{new}(\mathrm{dom}\,\pi)$ is the first variable not belonging to $\mathrm{dom}\,\pi$, in some standard ordering of the variables in $\mathcal{V}$. It is easy to see that $\mathcal{N}_{\pi\theta}$ is the set of long β-η-normal forms e that satisfy the typing $\pi \vdash e : \theta$. (The use of new imposes a canonical choice of a single normal form in each equivalence class under α-conversion.)

Each type-preserving variable map $\phi \in \pi \xrightarrow[\Sigma]{} \pi'$ induces a substitution $-/\phi$ that maps each $e \in \mathcal{M}_{\pi\theta}$ into $e/\phi \in \mathcal{M}_{\pi'\theta}$ and each $e \in \mathcal{N}_{\pi\theta}$ into $e/\phi \in \mathcal{N}_{\pi'\theta}$:

$$\begin{aligned} v/\phi &= \phi v \\ \big(\mathsf{appl}(e, e')\big)/\phi &= \mathsf{appl}(e/\phi, e'/\phi) \\ \big(\mathsf{abstr}(v, e)\big)/\phi &= \mathsf{abstr}(v', e/[\,\phi \mid v\colon v'\,]), \end{aligned}$$

where $v' = \mathsf{new}(\mathrm{dom}\,\pi')$.

It is easy to see that $e/(\phi\,;\phi') = (e/\phi)/\phi'$ and $e/I^{\Sigma}_{\pi} = e$. Thus we may complete our semantics by defining the functor $[\![\iota]\!]$:

$$[\![\iota]\!]\pi = \mathcal{N}_{\pi\iota} \qquad\qquad [\![\iota]\!](\phi \in \pi \xrightarrow[\Sigma]{} \pi')(e \in \mathcal{N}_{\pi\iota}) = e/\phi.$$

Next we define two families of transfer functions between sets of expressions and sets obtained from the meanings of types:

$$\tau_{\pi\theta} \in \mathcal{M}_{\pi\theta} \to [\![\theta]\!]\pi \qquad\qquad \mu_{\pi\theta} \in [\![\theta]\!]\pi \to \mathcal{N}_{\pi\theta},$$

by

$$\begin{aligned} \tau_{\pi\iota} e &= e \\ \tau_{\pi,\theta\to\theta'}(e \in \mathcal{M}_{\pi,\theta\to\theta'})\pi'\langle \phi \in \pi \xrightarrow[\Sigma]{} \pi', a \in [\![\theta]\!]\pi'\rangle &= \tau_{\pi'\theta'}\big(\mathsf{appl}(e/\phi, \mu_{\pi'\theta} a)\big) \\ \mu_{\pi\iota} e &= e \\ \mu_{\pi,\theta\to\theta'}(p \in [\![\theta]\!]\pi \Rightarrow [\![\theta']\!]\pi) &= \mathsf{abstr}\big(v, \mu_{\pi'\theta'}(p\pi'\langle J, \tau_{\pi'\theta} v\rangle)\big), \end{aligned}$$

where $v = \mathsf{new}(\mathrm{dom}\,\pi)$, $\pi' = [\,\pi \mid v\colon \theta\,]$, and $J \in \pi \xrightarrow[\Sigma]{} \pi'$ is the identity injection from $\mathrm{dom}\,\pi$ to $\mathrm{dom}\,\pi'$.

By induction on the structure of the type θ, one can show

Proposition 1 *If* $\phi \in \pi \xrightarrow[\Sigma]{} \pi'$ *then*

$$\begin{tikzcd} \mathcal{M}_{\pi\theta} \arrow[r, "\tau_{\pi\theta}"] \arrow[d, "-/\phi"] & {[\![\theta]\!]\pi} \arrow[d, "{[\![\theta]\!]\phi}"] \\ \mathcal{M}_{\pi'\theta} \arrow[r, "\tau_{\pi'\theta}"] & {[\![\theta]\!]\pi'} \end{tikzcd}$$

commutes in SET.

Now suppose we define the environment

$$\hat{\eta}_{\pi} = \lambda v \in \mathrm{dom}\,\pi.\ \tau_{\pi,\pi v} v \in [\![\pi]\!]\pi.$$

From the above proposition (taking ϕ to be J), and the action of $[\![\pi]\!]$ on morphisms of Σ, one can show

$$\hat{\eta}_{\pi'} = [\,[\![\pi]\!]J\hat{\eta}_{\pi} \mid v\colon \tau_{\pi'\theta} v\,],$$

where $v = \mathsf{new}(\mathrm{dom}\,\pi)$, $\pi' = [\,\pi \mid v\colon \theta\,]$, and $J \in \pi \xrightarrow[\Sigma]{} \pi'$ is the identity injection from $\mathrm{dom}\,\pi$ to $\mathrm{dom}\,\pi'$.

Finally, by simultaneous induction on the structure of the expression e, we obtain

Proposition 2

1. *If* $e \in \mathcal{M}_{\pi\theta}$ *then* $[\![e]\!]_{\pi\theta}\pi\hat{\eta}_\pi = \tau_{\pi\theta}e$.
2. *If* $e \in \mathcal{N}_{\pi\theta}$ *then* $\mu_{\pi\theta}([\![e]\!]_{\pi\theta}\pi\hat{\eta}_\pi) = e$.

Then, since an expression has the same meaning as its normal form, the second part of this proposition gives

Proposition 3 *If* $\pi \vdash e : \theta$ *then* $\mu_{\pi\theta}([\![e]\!]_{\pi\theta}\pi\hat{\eta}_\pi)$ *is the long* β-η*-normal form of* e.