An Interpretation for Typed Lambda Calculus: Kripke-Style Models and Complete Equation Calculus

Tomohiro Hoshi

06/25/2006

Motivation

The purpose of this presentation will be to give a Kripke-style semantics for typed lambda calculus and prove the completeness for an equation calculus that is complete with respect to the semantics.

- Syntax of Typed Lambda Calculus
- Equation Calculus (EC)
- Kripke Lambda Structure (KLS)
- Semantics for Typed Lambda Calculus based on KLS
- Soundness and Completeness Theorem of EC with respect to KLS

The presentation will be mainly based on "Kripke-style models for typed lambda calculus" by J. C. Mitchell and Eugenio Moggi.

1 Syntax of Typed Lambda Calculus

Type Assignment

A type assignment Γ is a finite set of fomulas of the form $x : \sigma$, with x a variable and σ a type, such that Γ does not contain two distinct formulas with the same variables x.

We write $\Gamma, x : \sigma$ for the type assingment $\Gamma \cup \{x : \sigma\}$, where x does not occur in Γ.

Terms

Terms are written in the form $\Gamma \triangleright M : \tau$, where Γ is a type assignment, M, a lambda term, and τ, a type. $\Gamma \triangleright M : \tau$ is read as "M has a type assignment τ relative to Γ".

Well-Typed Terms

Well-typed terms are defined as follows:

(*var*)

$$x : \tau \triangleright x : \tau$$

$(\rightarrow E)$

$$\frac{\Gamma \triangleright M : \sigma \rightarrow \tau, \Gamma \triangleright N : \sigma}{\Gamma \triangleright MN : \tau}$$

$(\rightarrow I)$

$$\frac{\Gamma, x : \sigma \triangleright M : \tau}{\Gamma \triangleright \lambda x : \sigma.M : \sigma \rightarrow \tau}$$

(*add var*)

$$\frac{\Gamma \triangleright M : \tau}{\Gamma, x : \sigma \triangleright M : \tau}$$

An easy inductiton shows that if $\Gamma \triangleright M : \sigma$ is well-typed, then then Γ must mention every free variable of M.

It is also easy to see closed terms come with the empty type assignment (or type assignments irrelevant to them). Thus, it is convenient to omit the type assignement in such cases. For instance, we write $\lambda x : \sigma.x$ for $\emptyset \triangleright \lambda x : \sigma.x : \sigma \rightarrow \sigma$.

Equations for Well-Typed Terms

With type assignments as a part of the syntactic formulation of terms, it is natural to write equations in the form

$$\Gamma \triangleright M = N : \tau$$

where we assume that $\Gamma \triangleright M : \tau$ and $\Gamma \triangleright N : \tau$ are both well-typed.

Also for a typographical reason, we omit types after equations when there is no confusion. Thus we often write $\Gamma \triangleright M = N$ instead of writing $\Gamma \triangleright M = N : \tau$

2 Equational Calculus for α, β, η-Equations

Axioms for reductions

(α) For $y \notin FV(M)$,

$$\Gamma \triangleright \lambda x : \sigma.M = \lambda y : \sigma[y/x].M$$

(β)

$$\Gamma \triangleright (\lambda x:\sigma.M)N = [N/x].M$$

(η) For $x \notin FV(M)$,

$$\Gamma \triangleright \lambda x:\sigma.Mx = M$$

Reflexivity Axiom

$$\Gamma \triangleright M = M : \sigma$$

Rules for Symmetricity and Transitivity

(*sym*)

$$\frac{\Gamma \triangleright M = N : \sigma}{\Gamma \triangleright N = M : \sigma}$$

(*trans*)

$$\frac{\Gamma \triangleright M = N : \sigma, \Gamma \triangleright N = P : \sigma}{\Gamma \triangleright M = P : \sigma}$$

Rules for Congruence with respect to Application and lambda abstraction

(*cong*)

$$\frac{\Gamma \triangleright M_1 = M_2 : \sigma \to \tau, \Gamma \triangleright N_1 = N_2 : \sigma}{\Gamma \triangleright M_1 N_1 = M_2 N_2 : \tau}$$

(ξ)

$$\frac{\Gamma, x:\sigma \triangleright M = N : \tau}{\Gamma \triangleright \lambda x:\sigma.M = \lambda x:\sigma.N : \sigma \to \tau}$$

Rule for Additional Type Assignments

(*add var*)

$$\frac{\Gamma \triangleright M = N : \tau}{\Gamma, x:\sigma \triangleright M = N : \tau}$$

We can extend this calculus by adding boolean operators and quantifiers in a standard way.

Provability

If the equation $\Gamma \triangleright M = N : \sigma$ is provable from the equations in a set E, we write

$$E \vdash \Gamma \triangleright M = N : \sigma.$$

3 Kripke Lambda Structure

Kripke Applicative Structure (KAS)

A *Kripke applicative structure* A is a tuple:

$$\left\langle W, \leq, \{A_w^\sigma\}, \{App_w^{\sigma,\tau}\}, \{i_{w,w'}^\sigma\}\right\rangle,$$

that satisfies the conditions below.

- W is a set, which may be thought of as a set of "possible worlds", or in our case a set of type assignments.
- $\leq$ is a partial order on W, which may be thought of as accessibility relations between worlds.
- a family$\{A_w^\sigma\}$ is a family of sets indexed by a type σ and a world w, which may be thought of as a set of *meanings* assigned at w for terms typed as σ.
- a family $\{App_w^{\sigma,\tau}\}$ is a family of maps $App_w^{\sigma,\tau} : A_w^{\sigma\to\tau} \times A_w^\sigma \to A_w^\tau$ indexed by a pair of types σ, τ and a world w. With the restrictions below, this function can be thought of as the function that yields the meaning of the result (a term typed as τ) of application of a term typed as $\sigma \to \tau$ to a term typed as σ.
- a family $\{i_{w,w'}^\sigma\}$ is a family of maps $i_{w,w'}^\sigma : A_w^\sigma \to A_{w'}^\sigma$ indexed by a type σ and a pair of worlds w, w' such that $w \leq w'$. With the restriction below, this may be thougth of as working as correlating, or identifying, a meaning for a term at w with a meaning for the term at w'.

Conditions for KAS

Identity Condition on $i_{w,w}^\sigma$

(id)

$$i_{w,w}^\sigma : A_w^\sigma \to A_w^\sigma \ \textit{is the identity.}$$

Composition Condition on $i_{w,w'}^\sigma$

$(comp)$

$$i_{w',w''}^\sigma \circ i_{w,w'}^\sigma = i_{w,w''}^\sigma \ \textit{for all}\ w \leq w' \leq w''.$$

Commutability Condition on i and App

$(comm)$

$$i_{w,w'}^\sigma \left(App_w^{\sigma,\tau}(f,a)\right) = App_{w'}^{\sigma,\tau}\left(i_{w,w'}^\sigma(f), i_{w,w'}^\sigma(a)\right).$$

Kripke Lambda Structure (KLS)

A KAS may still not be a good candidate for an interpretation of typed lambda terms. There are two reasons:

1. There may not be enough elements at each world. If $A_w^{\sigma\to\sigma}$ is empty for all w, then we would not be able to assign a meaning to, say, the identity funciton $\lambda x : \sigma.x$.
2. There may be two distinct elements of the same functional type which have the same functional behavior, in which case the extensional nature of functions would not be represented. For instance, the meaning of a term $\lambda x : \sigma.M$ may not be determined uniquely.

For these two reasons, we put more constraint on a KAS to obtain an interpretation for typed lambda calculus.

Additional Conditions for KLS

A Kripke lambda structure is a KAS that satisfies the following two conditions: the extensionality condition and the condition for having combinators.

Extensionality Condition

Let A be a KAS and $w \in W$. Then, for all $f, g \in A_w^{\sigma\to\tau}$,

$$f = g \quad whenever \quad \forall w' \geq w . \forall a \in A_{w'}^{\sigma} . (i_{w,w'}^{\sigma\to\tau} f)a = (i_{w,w'}^{\sigma\to\tau} g)a.$$

Conditions for having Combanators

Global Elements

A global element $a : \sigma$ of a KAS A is a mapping that assign to a world w an element satisfying:

1. $a_w \in A_w^{\sigma}$
2. whenever $w \leq w'$, we have $i_{w,w'}^{\sigma} a_w = a_{w'}$

Having combinators

Let A be a K*AS*. Then, for every type ρ, σ, τ, there exist global elements K and S such that:

$$App_w^{\tau,\sigma}(App_w^{\sigma,\tau\to\sigma}(Kw, a), b) = a.$$

for every w in W of A, *every a,* in A_w^{σ}; and *every* b in A_w^{τ}, (for short, $K_w ab{=}a$) and:

$$App_w^{\rho;\tau}(App_w^{\rho\to\sigma,\rho\to\tau}(App_w^{\rho\to\sigma\to\tau,(\rho\to\sigma)\to\rho\to\tau}(Sw,s),t),u) = App_w^{\sigma;\tau}(App_w^{\rho;\sigma\to\tau}(s,u), App_w^{\rho;\sigma}(t,u))$$

for every w in W of A; *every* s, in $A_w^{\rho\to\sigma\to\tau}$; *every* t in $A_w^{\rho\sigma}$; and every u in A_w^{ρ} (for short, $S_w stu{=}su(tu)$.

4 Semantics for Typed Lambda Calculus based on KLS

Now we give the semantics for typed lambda calculus based on KLS.

Environment (Variable Assignment)

An environment η (with respect to a KLS A) is a partial mapping from variables and worlds to elements of A such that:

$$If \quad \eta xw \in A_w^\sigma \quad and \quad w' \geq w, \quad then \quad \eta xw' = i_{w,w'}^\sigma(\eta xw).$$

We write $\eta[a/x]$ for the environment identitcal to η on variables except for x, which, for all $w' \geq w$, satisfies:

$$(\eta[a/x])xw' = i_{w,w'}^\sigma a.$$

Satisfaction of Type Assignments with respect to Worlds and Environments

We write $w \models \Gamma[\eta]$, if

$$\eta xw \in A_w^\sigma \quad for \quad all \quad x : \sigma \in \Gamma.$$

Notice that, if $w \models \Gamma[\eta]$, then $w' \models \Gamma[\eta]$ for all $w' \geq w$.

Semantics for Typed Lambda Terms

For every KLS A and evironment η such that $w \models \Gamma[\eta]$, we define the interpretation $||\Gamma \triangleright M : \sigma||\eta w$ of a well-typed term$\Gamma \triangleright M : \sigma$ with respect to η at a world w in the following way:

$$\|\Gamma \triangleright x : \sigma\|\, \eta w = \eta xw$$

$$\|\Gamma \triangleright MN : \tau\|\, \eta w = App_w^{\sigma;\tau}(\|\Gamma \triangleright M : \sigma \to \tau\|\, \eta w)(\|\Gamma \triangleright N : \sigma\|\, \eta w)$$

$$\|\Gamma \triangleright \lambda x : \sigma.M : \sigma \to \tau\|\, \eta w = the \quad unique \quad d \in A_w^{\sigma\to\tau}$$

$$such\ that\ for\ all\ a \in A^{\sigma}_{w'}\ and\ w' \geq w,$$
$$App^{\sigma,\tau}_{w'}(i^{\sigma\to\tau}_{w,w'}d)a = \|\Gamma, x : \sigma \triangleright M : \tau\|\, \eta[a/x]w'$$

We can prove the existence and unqueness of d. The key is the extensionality conditioin and the condition for combinators. For the purpose of the presentation, we do not look at the detial. Instead we refer interested hearers to by *The Lambda Calculus: Its Syntax and Semantics*, H.P. Berendregt in 1984.

Satisfaction for Equations

We write $w \models (\Gamma \triangleright M = N : \sigma)[\eta]$, if,

$$whenever\ w \models \Gamma[\eta],$$

$$\|\Gamma \triangleright M : \sigma\|\, \eta w = \|\Gamma \triangleright N : \sigma\|\, \eta w.$$

We also write $A \models \Gamma \triangleright M = N : \sigma$ for a KLS A, if, for every w in A and every environment η with respect to A, $w \models (\Gamma \triangleright M = N : \sigma)[\eta]$.

We can extend the notion of satisfaction based on this by adding boolean operators and quantifiers in a standard way.

5 Soundness and Completeness Theorems

Lemma 1 (*Transition Lemma*)hspace1em Let A be a KLS and η an environment satisfying Γ at w. Then for every $w \geq w'$, we have

$$\|\Gamma \triangleright M : \sigma\|\, \eta w' = i^{\sigma}_{w,w'}(\|\Gamma \triangleright M : \sigma\|\, \eta w).$$

Proof: Simple induction on the complexity of M.

Lemma 2 (*Substitution Lemma*)hspace1em Let A be a KLS and η an environment satisfying Γ at w. For any well-typed term, $\Gamma \triangleright N : \sigma$ and $\Gamma, x : \sigma \triangleright M : \tau$, we have

$$\|\Gamma \triangleright [N/x]M : \tau\|\, \eta w = \|\Gamma, x : \sigma \triangleright M : \tau\|\, (\eta[\|\Gamma \triangleright N : \sigma\|\, \eta w/x])w.$$

Proof: Simple induction on the complexity of M.

Theorem 3 (*Soundess Theorem*)hspace1em Let E be a set of well-typed equations. If $E \vdash \Gamma \triangleright M = N : \sigma$, then every model satisfying E also satisfies $\Gamma \triangleright M = N : \sigma$.

Proof: Easily follows from the previous lemmas.

Now, we prove the completeness, as usual, by constructing a so-called canonical model.

Theorem 4 (*Completeness Theorem*)hspace1em Let E be a set of well-typed equations closed under $\vdash$. Then, there is a KLS A such that $A \models \Gamma \triangleright M = N : \sigma$ iff $\Gamma \triangleright M = N : \sigma$ is in E.

Proof: We construct a KLM $A{=}\langle W, \leq, \{A^\sigma_w\}, \{App^{\sigma,\tau}_w\}, \{i^\sigma_{w,w'}\}\rangle$ in the following way:

- W is the partially ordered set of finite type assignments Γ ordered by inclusion. In what follows, we will write Γ for an arbitrary element of W.
- A^σ_Γ is the set of all $[\Gamma \triangleright M : \sigma]$, where $\Gamma \triangleright\ M : \sigma$ is well-typed, and $[\Gamma \triangleright\ M : \sigma]{=}\{\Gamma \triangleright\ N : \sigma \mid E \vdash \Gamma \triangleright M = N : \sigma\}$.
- $App^{\sigma,\tau}_\Gamma([\Gamma \triangleright M : \sigma \to \tau], [\Gamma \triangleright N : \sigma]) = [\Gamma \triangleright MN : \tau]$.
- $i^\sigma_{\Gamma,\Gamma'}([\Gamma \triangleright M : \sigma]) = [\Gamma' \triangleright M\sigma]$ for $\Gamma \subseteq \Gamma'$.

Here, it is easy to see that A is at least a KAS. So we would like to check if it satisfies the extensionality and the combinator. It is easy to see the combinator condition is satisfied since we have, for instance,

$$K = [\lambda x : \sigma.\lambda y : \tau.x]$$

$$S = [\lambda x : \sigma.\lambda y : \tau.\lambda z : \rho.xz(yz)]$$

Thus, we now prove the extensionality. Suppose that $[\Gamma \triangleright\ M : \sigma \to \tau]$ and $[\Gamma \triangleright\ N : \sigma \to \tau]$ have the same functional behavior, i.e., for all $\Gamma' \geq \Gamma$ and $\Gamma' \triangleright\ P : \sigma$, we have

$$[\Gamma' \triangleright MP : \tau] = [\Gamma' \triangleright NP : \tau].$$

Then, in particular, for $\Gamma' \equiv \Gamma, x : \sigma$ with x not in Γ, we have

$$[\Gamma, x : \sigma \triangleright Mx : \tau] = [\Gamma, x : \sigma \triangleright Nx : \tau]$$

and so by rule (ξ) and axiom (η), we have $[\Gamma \triangleright M : \sigma \to \tau] = [\Gamma \triangleright N : \sigma \to \tau]$. Thus, A is a KLS.

Now, we show that this MLS A is the model that is claimed in the theorem, i.e. A satisfies precisely the formulas in E.

($\Rightarrow$)

Given a type assignemnt Γ, we may define an environment η as follows:

$$\eta x \Gamma = \begin{cases} [\Gamma' \triangleright x : \sigma] & if\ \ x : \sigma \in \Gamma \subseteq \Gamma' \\ undefined & otherwise \end{cases}$$

A straightforward induction on the complexity of terms shows that for any Γ, we have

$$||\Gamma \triangleright M : \sigma||\eta\Gamma = [\Gamma \triangleright M : \sigma].$$

In particular, if A saitsfies an equation $\Gamma \triangleright M = N : \sigma$, we have $\Gamma \models \Gamma[\eta]$ by the construction of η and so

$$[\Gamma \triangleright M : \sigma] = [\Gamma \triangleright N : \sigma].$$

This applies to every Γ and every equation. Therefore, every equation satisfied by A must be provable from E.

($\Leftarrow$)

First, notice that the closed term equation

$$\emptyset \triangleright \lambda x_1 : \sigma_1 ... \lambda x_k : \sigma_k.M = \lambda x_1 : \sigma_1 ... \lambda x_k : \sigma_k.N.$$

is provable from

$$x_1 : \sigma_1, ..., x_k : \sigma_k \triangleright M = N$$

and vice versa by the successive application of $(cong)$ or (ξ). Therefore, by the soundness, without loss of generality, we can only look at closed terms.

For any closed term equation $\emptyset \triangleright M = N : \tau$, we have

$$E \vdash \Gamma \triangleright M = N : \tau$$

for any Γ, by the successive applicatino of (*add var*). Therefore, for every world Γ of A, the two equivalence classes $[\Gamma \triangleright M : \tau]$ and $[\Gamma \triangleright N : \tau]$ are identical. Here, we have seen above that $||\Gamma \triangleright M : \sigma||\eta\Gamma = [\Gamma \triangleright M : \sigma]$. Since the meaning of closed terms is not affected by an environment, it follows that, for all environment η', $||\emptyset \triangleright M : \sigma||\eta\emptyset = [\emptyset \triangleright M : \sigma]$. Then, by the definition of $i^{\sigma}_{\emptyset,\Gamma}$, the meaning of $\emptyset \triangleright M : \tau$ in any environment at any world Γ is $[\Gamma \triangleright M : \tau]$, and similarly for. $\emptyset \triangleright N : \tau$. Therefore, A satisfies $\emptyset \triangleright M = N : \tau$. This completes the proof.