

MPRI, Typage

Didier Rémy
(With course material from François Pottier)

December 11, 2014



Plan of the course

Introduction

Simply-typed λ -calculus

Polymorphism and System F

Type reconstruction

Existential types

Overloading

Logical relations

Logical relations and parametricity

Contents

- Introduction
- Normalization of stlc
- Observational equivalence in stlc
- Logical relations in stlc
- Logical relations in F

What are they?

So far, most proofs involving terms have been by induction on the structure of *terms*.

Logical relations are relations between well-typed terms defined inductively on the structure of *types*. They allow proofs between terms by induction on the structure of *types*.

Unary relations

- Unary relations are predicates on expressions
- They can be used to prove type safety and strong normalisation

Binary relations

- Binary relations relates two expressions of related types.
- They can be used to prove equivalence of programs and non-interference properties.

Logical relations are a common proof method for programming languages.

Contents

- Introduction
- Normalization of stlc
- Observational equivalence in stlc
- Logical relations in stlc
- Logical relations in F



Normalization of simply-typed λ -calculus

Types usually ensure termination of programs—as long as neither types nor terms contain any form of recursion.



Normalization of simply-typed λ -calculus

Types usually ensure termination of programs—as long as neither types nor terms contain any form of recursion.

Even if one wishes to add recursion explicitly later on, it is an important property of the design that non-termination is originating from the constructions introduced especially for recursion and could not occur without them.



Normalization of simply-typed λ -calculus

Types usually ensure termination of programs—as long as neither types nor terms contain any form of recursion.

Even if one wishes to add recursion explicitly later on, it is an important property of the design that non-termination is originating from the constructions introduced especially for recursion and could not occur without them.

The simply-typed λ -calculus is also lifted at the level of types in richer type systems such as System F^ω ; then, the decidability of type-equality depends on the termination of the reduction at the type level.



Normalization of simply-typed λ -calculus

Types usually ensure termination of programs—as long as neither types nor terms contain any form of recursion.

Even if one wishes to add recursion explicitly later on, it is an important property of the design that non-termination is originating from the constructions introduced especially for recursion and could not occur without them.

The simply-typed λ -calculus is also lifted at the level of types in richer type systems such as System F^ω ; then, the decidability of type-equality depends on the termination of the reduction at the type level.

The proof of termination for the simply-typed λ -calculus is a simple and illustrative use of logical relations.

Notice however, that our simply-typed λ -calculus is equipped with a call-by-value semantics. Proofs of termination are usually done with a strong evaluation strategy where reduction can occur in any context.



Normalization

Proving termination of reduction in fragments of the λ -calculus is often a difficult task because reduction may create new redexes or duplicate existing ones.

Hence the size of terms may grow (much) larger during reduction. The difficulty is to find some underlying structure that decreases.

We follow the proof schema of Pierce [2002], which is a modern presentation in a call-by-value setting of an older proof by Hindley and Seldin [1986]. The proof method is due to [Tait, 1967].

Calculus

Take the call-by-value simply-typed λ -calculus with primitive booleans and conditional.

Write B the type of booleans and tt and ff for *true* and *false*.

We define $\mathcal{V}(\tau)$ and $\mathcal{E}(\tau)$ the subsets of closed values and closed expressions of type τ by [induction on types](#) as follows:

$$\mathcal{V}(B) \triangleq \{\text{tt}, \text{ff}\}$$

$$\mathcal{V}(\tau_1 \rightarrow \tau_2) \triangleq \{\lambda x:\tau_1. M \mid \lambda x:\tau_1. M : \tau_1 \rightarrow \tau_2 \wedge \forall V \in \mathcal{V}(\tau_1), M V \in \mathcal{E}(\tau_2)\}$$

$$\mathcal{E}(\tau) \triangleq \{M \mid M : \tau \wedge \exists V \in \mathcal{V}(\tau), M \longrightarrow^* V\}$$

The goal is to show that any closed expression of type τ is in $\mathcal{E}(\tau)$.

Remarks

$\mathcal{V}(\tau) \subseteq \mathcal{E}(\tau)$ —by definition.

$\mathcal{E}(\tau)$ is closed by inverse reduction—by definition, *i.e.*

If $M : \tau$ and $M \longrightarrow N$ and $N \in \mathcal{E}(\tau)$ then $M \in \mathcal{E}(\tau)$.

Problem

We wish to show that every closed term of type τ is in $\mathcal{E}(\tau)$

- Proof by induction on the typing derivation.
- Problem with abstraction: the premisses are not closed.

We need to strengthen the hypothesis, *i.e.* also give a semantics to open terms.

- The semantics of open terms can be given by abstracting over the semantics of their free variables.

Generalize the definition to open terms

We define a semantic judgment for open terms $\Gamma \vDash M : \tau$ so that $\Gamma \vdash M : \tau$ implies $\Gamma \vDash M : \tau$ and $\emptyset \vDash M : \tau$ means $M \in \mathcal{E}(\tau)$.

We interpret free type variables of type τ as closed values in $\mathcal{V}(\tau)$.

We interpret environments Γ as mappings γ from type variables to *closed values*:

We write $\gamma \in \mathcal{G}(\Gamma)$ to mean $\text{dom}(\gamma) = \text{dom}(\Gamma)$ and $\gamma(x) \in \mathcal{V}(\tau)$ for all $x : \tau \in \Gamma$.

$$\Gamma \vDash M : \tau \stackrel{\text{def}}{\iff} \forall \gamma \in \mathcal{G}(\Gamma), \gamma(M) \in \mathcal{E}(\tau)$$

Fundamental Lemma

Theorem (fundamental lemma)

If $\Gamma \vdash M : \tau$ then $\Gamma \vDash M : \tau$.

Corollary (termination of well-typed terms):

If $\emptyset \vdash M : \tau$ then $M \in \mathcal{E}(\tau)$.

That is, closed well-typed terms of type τ evaluates to values of type τ .

Proof by induction on the typing derivation

Routine cases

Case $\Gamma \vdash tt : B$ or $\Gamma \vdash ff : B$: by definition, $tt, ff \in \mathcal{V}(B)$ and $\mathcal{V}(B) \subseteq \mathcal{E}(B)$.

Case $\Gamma \vdash x : \tau$: $\gamma \in \mathcal{G}(\Gamma)$, thus $\gamma(x) \in \mathcal{V}(\tau) \subseteq \mathcal{E}(\tau)$

Case $\Gamma \vdash M_1 M_2 : \tau$:

By inversion, $\Gamma \vdash M_1 : \tau_2 \rightarrow \tau$ and $\Gamma \vdash M_2 : \tau_2$.

Let $\gamma \in \mathcal{G}(\Gamma)$. We have $\gamma(M_1 M_2) = (\gamma M_1) (\gamma M_2)$.

By IH, we have $\Gamma \vDash M_1 : \tau_2 \rightarrow \tau$ and $\Gamma \vDash M_2 : \tau_2$.

Thus $\gamma M_1 \in \mathcal{E}(\tau_2 \rightarrow \tau)$ **(1)** and $\gamma M_2 \in \mathcal{E}(\tau_2)$ **(2)**.

By (2), there exists $V \in \mathcal{V}(\tau_2)$ such that $\gamma M_2 \longrightarrow^* V$.

Thus $(\gamma M_1) (\gamma M_2) \rightsquigarrow (\gamma M_1) V \in \mathcal{E}(\tau)$ by (1).

Then, $(\gamma M_1) (\gamma M_2) \in \mathcal{E}(\tau)$, by closure by inverse reduction.

Case $\Gamma \vdash \text{if } M \text{ then } M_1 \text{ else } M_2 : \tau$: By cases on the evaluation of γM .



Proof by induction on the typing derivation

The interesting case

Case $\Gamma \vdash \lambda x:\tau_1. M : \tau_1 \rightarrow \tau$:

Assume $\gamma \in \mathcal{G}(\Gamma)$.

We must show that $\gamma(\lambda x:\tau_1. M) \in \mathcal{E}(\tau_1 \rightarrow \tau)$ (1)

That is, $\lambda x:\tau_1. \gamma M \in \mathcal{V}(\tau_1 \rightarrow \tau)$ (we may assume $x \notin \text{dom}(\gamma)$ w.l.o.g.)

Let $V \in \mathcal{V}(\tau_1)$, it suffices to show $(\lambda x:\tau_1. \gamma M) V \in \mathcal{E}(\tau)$ (2).

We have $(\lambda x:\tau_1. \gamma M) V \longrightarrow (\gamma M)[x \mapsto V] = \gamma' M$

where γ' is $\gamma[x \mapsto V] \in \mathcal{G}(\Gamma, x:\tau_1)$ (3)

Since $\Gamma, x:\tau_1 \vdash M : \tau$, we have $\Gamma, x:\tau_1 \models M : \tau$ by IH. Therefore by (3), we have $\gamma' M \in \mathcal{E}(\tau)$. Since $\mathcal{E}(\tau)$ is closed by inverse reduction, this proves (2) which finishes the proof of (1).

Variations

We have shown both *termination* and *type soundness*, simultaneously.

Termination would not hold if we had a fix point. But type soundness would still hold.

The proof may be modified by choosing:

$$\mathcal{E}(\tau) = \{M : \tau \mid \forall N, M \longrightarrow^* N \implies N \in \mathcal{V}(\tau) \vee \exists N', N \longrightarrow N'\}$$

Exercise

Show type soundness with this semantics.



Contents

- Introduction
- Normalization of stlc
- Observational equivalence in stlc
- Logical relations in stlc
- Logical relations in F

(Bibliography)

Mostly following Bob Harper's course notes *Practical foundations for programming languages* [Harper, 2012].

See also

- *Types, Abstraction and Parametric Polymorphism* [Reynolds, 1983]
- *Parametric Polymorphism and Operational Equivalence* [Pitts, 2000].
- *Theorems for free!* [Wadler, 1989].

We assume a call-by-name semantics for generality of the presentation.

When are two programs equivalent

$M \longrightarrow^* N$?

$M \longrightarrow^* V$ and $N \longrightarrow^* V$?

But what if M and N are functions?

Aren't $\lambda x. (x + x)$ and $\lambda x. 2 * x$ equivalent?

Idea two functions are observationally equivalent if when applied to *equivalent arguments*, they lead to observationally *equivalent results*.

Are we general enough?

Observational equivalence

We can only *observe* the behavior of full *programs*, *i.e.* closed terms of some computation type, such as B (the only one so far).

If $M : B$ and $N : B$, then $M \simeq N$ iff there exists V such that $M \longrightarrow^* V$ and $N \longrightarrow^* V$. (Call $M \simeq N$ *behavioral equivalence*.)

To compare programs at other types, we place them in *closing* arbitrary contexts.

Definition (observational equivalence)

$$\Gamma \vdash M \cong N : \tau \stackrel{\Delta}{=} \forall C : (\Gamma \triangleright \tau) \rightsquigarrow (\emptyset \triangleright B), C[M] \simeq C[N]$$

Typing of contexts

$$C : (\Gamma \triangleright \tau) \rightsquigarrow (\Delta \triangleright \sigma) \iff (\forall M, \Gamma \vdash M : \tau \implies \Delta \vdash C[M] : \sigma)$$

There is an equivalent definition given by a set of typing rules. This is useful to proof some properties by induction on the derivations.

We write $M \cong_{\tau} N$ for $\emptyset \vdash M \cong N : \tau$



Observational equivalence

Observational equivalence is the coarsest consistent congruence, where:

\equiv is consistent if $\emptyset \vdash M \equiv N : B$ implies $M \simeq N$.

\equiv is a congruence if it is an equivalence and is closed by context, *i.e.*

$$\Gamma \vdash M \equiv N : \tau \wedge \mathcal{C} : (\Gamma \triangleright \tau) \rightsquigarrow (\Delta \triangleright \sigma) \implies \Delta \vdash \mathcal{C}[M] \equiv \mathcal{C}[N] : \sigma$$

Consistent: by definition, using the empty context.

Congruence: by compositionality of contexts.

Largest: Assume \equiv is a consistent congruence. Assume $\Gamma \vdash M \equiv N : \tau$ holds and show that $\Gamma \vdash M \simeq N : \tau$ holds (1).

Let $\mathcal{C} : (\Gamma \triangleright \tau) \rightsquigarrow (\emptyset \triangleright B)$ (2). We must show that $\mathcal{C}[M] \simeq \mathcal{C}[N]$.

This follows by consistency applied to $\Gamma \vdash \mathcal{C}[M] \equiv \mathcal{C}[N] : B$ which follows by congruence from (1) and (2).

Contents

- Introduction
- Normalization of stlc
- Observational equivalence in stlc
- Logical relations in stlc
- Logical relations in F

Problem with Observational Equivalence

Problems

- Observational equivalence is too difficult to test.
- Because of quantification over all contexts (too many for testing).
- But many contexts will do the same experiment.

Solution

We take advantage of types to reduce the number of experiments.

- Defining/testing the equivalence on base types.
- Propagating the definition mechanically at other types.

Logical relations provide the infrastructure for conducting such proofs.

Logical equivalence for closed terms

We inductively define $M \sim_{\tau} M'$ on closed terms of type τ by induction on τ :

- $M \sim_{\mathbf{B}} M'$ iff $M \simeq M'$
- $M \sim_{\tau_1 \rightarrow \tau_2} M'$ iff $\forall M_1, M'_1, M \sim_{\tau_1} M' \implies M M_1 \sim_{\tau_2} M' M'_1$

Lemma

Logical equivalence is symmetric and transitive (at any given type).

Note

Reflexivity is not obvious at all.

Logical equivalence for closed terms

Proof by induction on type τ

Case τ is B for values: the result is immediate.

Case τ is $\tau_1 \rightarrow \tau_2$: By IH, symmetry and transitivity holds at types τ_1 and τ_2 .

For symmetry, assume $M \sim_{\tau} M'$ (H), we must show $M' \sim_{\tau} M$. Assume $M_1 \sim_{\tau_1} M'_1$. We must show $M' M_1 \sim_{\tau_2} M M'_1$ (C). We have $M'_1 \sim_{\tau_1} M_1$ by symmetry at τ_1 . By (H), we have $M M'_1 \sim_{\tau_2} M' M_1$ and (C) follows by symmetry at type τ_2 .

For transitivity, assume $M \sim_{\tau} M'$ (H1) and $M' \sim_{\tau} M''$ (H2). To show $M \sim_{\tau} M''$, we assume $N \sim_{\tau_1} N''$ and show $M N \sim_{\tau_2} M'' N''$ (C).

By (H1), we have $M N \sim_{\tau_2} M' N''$ (C1).

By symmetry and transitivity at type τ_1 , we have $N'' \sim_{\tau_1} N'$. (Remark)

By (H2), we have $M' N'' \sim_{\tau_2} M'' N''$ (C2).

(C) follows by transitivity of (C1) and (C2) at type τ_2 .

Properties of logical equivalence

Closure by inverse reduction

Assume that $N : \tau$ and $M \sim_{\tau} M'$.

If $N \longrightarrow^* M$ and $N' \longrightarrow^* M'$ then $N \sim_{\tau} N'$.

The proof is by induction on τ .

(We show it for a single reduction step, e.g. on the left-hand side)

Case τ is B: By closure of behavioral equivalence \simeq by inverse reduction.

Case τ is $\tau_1 \rightarrow \tau_2$: To show $N \sim_{\tau} M'$ we assume $M_1 \sim_{\tau_1} M'_1$ and show $N M_1 \sim_{\tau_2} M' M'_1$ (1).

From $M \sim_{\tau} M'$, we have $M M_1 \sim_{\tau_2} M' M'_1$. The conclusion (1) then follows by IH at type τ_2 , since we have $N M_1 \longrightarrow M M_1$ as a consequence of the assumption $N \longrightarrow M$.

Consistency If $M \sim_{\text{B}} M'$, then $M \simeq M'$

(Obvious, by definition.)

Logical equivalence for open terms

When $\Gamma \vdash M : \tau$ and $\Gamma \vdash M' : \tau$, we wish to define a judgment $\Gamma \vdash M \sim M' : \tau$ to mean that the open terms M and M' are equivalent at type τ .

We write $\gamma \sim_{\Gamma} \gamma'$ to mean that γ and γ' are two substitutions of domain $\text{dom}(\Gamma)$ such that for all $x : \tau \in \text{dom}(\Gamma)$, we have $\gamma(x) \sim_{\tau} \gamma'(x)$

Definition

$$\Gamma \vdash M \sim M' : \tau \iff \forall \gamma, \gamma', \gamma \sim_{\Gamma} \gamma' \implies \gamma(M) \sim_{\tau} \gamma'(M')$$

We write $M \sim_{\tau} N$ for $\emptyset \vdash M \sim N : \tau$

Immediate properties

Open logical equivalence is symmetric and transitive.

(Proof is immediate by the definition and the symmetry and transitivity of closed logical equivalence.)

Fundamental lemma of logical equivalence

Reflexivity If $\Gamma \vdash M : \tau$, then $\Gamma \vdash M \sim M : \tau$.

Proof Assume $\Gamma \vdash M : \tau$ (1) and $\gamma \sim_{\Gamma} \gamma'$ (2). We must show $\gamma M \sim_{\tau} \gamma' M$. The proof is by induction on the typing derivation.

Case M is $\lambda x:\tau_1. N$ and τ is $\tau_1 \rightarrow \tau_2$ with $x \# \gamma, \gamma'$:

We show $\lambda x:\tau_1. \gamma N \sim_{\tau_1 \rightarrow \tau_2} \lambda x:\tau_1. \gamma' N$. Assume $M_1 \sim_{\tau_1} M'_1$ (3).

We must show $(\lambda x:\tau_1. \gamma N) M_1 \sim_{\tau_2} (\lambda x:\tau_1. \gamma' N) M'_1$.

By inverse reduction, it suffices to show

$\gamma(N)[x \mapsto M_1] \sim_{\tau_2} \gamma'(N)[x \mapsto M'_1]$, i.e.

$\gamma_1(N) \sim_{\tau_2} \gamma'_1(N)$ where γ_1 is $(\gamma[x \mapsto M_1])$ and γ'_1 is $(\gamma'[x \mapsto M'_1])$ (4).

We have $\gamma_1 \sim_{\Gamma, x:\tau_1} \gamma'_1$ (5) from (2) and (3).

By inversion of typing applied to (1), we have $\Gamma, x:\tau_1 \vdash N : \tau_2$.

Thus (4) follows by induction hypothesis applied with (5).

Properties of logical equivalence

Proof (continued) Assume $\Gamma \vdash M : \tau$ and $\gamma \sim_{\Gamma} \gamma'$. We must show $\gamma(M) \sim_{\tau} \gamma'(M)$. The proof is by induction on the typing derivation.

Case M is tt or ff and τ is B: Since M is closed it suffices to show $M \sim_{\mathbf{B}} M$ which holds by reflexivity of $\sim_{\mathbf{B}}$, i.e. of behavioral equivalence \simeq .

Case M is $M_1 M_2$: By induction hypothesis and the fact that substitution distributes over term application.

Case M is x : Immediate.

Properties of logical equivalence

Proof (continued)

Case M is if N then N_1 else N_2 : By induction applied to $\Gamma \vdash N : B$, we have $\Gamma \vdash N \sim N : B$. Thus $\gamma N \sim_B \gamma' N$. By consistency, we have $\gamma N \simeq \gamma' N$. We reason by cases on the evaluation of γN .

If $\gamma N \longrightarrow^* \text{tt}$ then so does $\gamma' N$; then $\gamma M \longrightarrow^* \gamma N_1$ and $\gamma' M \longrightarrow^* \gamma' N_1$. We have $\Gamma \vdash N_1 : \tau$ by inversion of typing. By IH, we have $\gamma N_1 \sim_\tau \gamma' N_1$. By inverse reduction, we get $\gamma M \longrightarrow^* \gamma' M$.

Otherwise, $\gamma N \longrightarrow^* \text{ff}$, and we proceed symmetrically.

Properties of logical relations

Corollary (equivalence) Open logical relation is an equivalence relation

Corollary (Termination) If $M : B$ then the evaluation of M terminates.

Proof: $M : B$ implies $M \sim_B M$ which implies $M \simeq M$, and, in turn, implies that M evaluates to either tt or ff.

Properties of logical equivalence

Logical equivalence is a congruence

If $\Gamma \vdash M \sim M' : \tau$ and $\mathcal{C} : (\Gamma \triangleright \tau) \rightsquigarrow (\Delta \triangleright \sigma)$, then
 $\Delta \vdash \mathcal{C}[M] \sim \mathcal{C}[M'] : \sigma$.

Proof By induction on the proof of $\mathcal{C} : (\Gamma \triangleright \tau) \rightsquigarrow (\Delta \triangleright \sigma)$.

Similar to the proof of reflexivity. (We need a definition of context typing derivations by a set of typing rules to be able to reason by induction on the typing derivation.)

Corollary Logical equivalence implies observational equivalence.

If $\Gamma \vdash M \sim M' : \tau$ then $\Gamma \vdash M \cong M' : \tau$.

Proof: Logical equivalence is a consistent congruence, hence included in observational equivalence which is the coarsest such relation.

Properties of logical equivalence

Lemma

Observational equivalence of closed terms implies logical equivalence.

If $M \cong_{\tau} M'$ then $M' \sim_{\tau} M'$.

Proof by induction on τ .

Case τ is B: With the empty context, we have $M \simeq M'$, hence $M \simeq M'$.

Case τ is $\tau_1 \rightarrow \tau_2$: By congruence of observational equivalence. To show $M \sim_{\tau} M'$, we assume $M_1 \sim_{\tau_1} M'_1$ (1) and show $M M_1 \sim_{\tau_2} M' M_1$. By IH, it suffices to show $M M_1 \cong_{\tau_2} M' M_1$. This follows by congruence, from the hypothesis $M \cong_{\tau} M'$ and $M_1 \cong_{\tau_1} M'_1$ which follows from (1) by the previous lemma.

Properties of logical equivalence

Corollary (Value arguments)

To show $M \sim_{\tau_1 \rightarrow \tau_2} M'$, it suffices to show that $M V \sim_{\tau_2} M' V'$ for all values V and V' such that $V \sim_{\tau_1} V'$.

Proof

Assume $N \sim_{\tau_1} N'$.

There exists V and V' such that $N \longrightarrow^* V$ and $N' \longrightarrow^* V'$.

It suffices to show that $M V \sim_{\tau_2} M' V'$ (H) implies $M N \sim_{\tau_2} M' N'$ (1).

We have $N \sim_{\tau_1} V$ from $N \longrightarrow^* V$.

Then $M N \sim_{\tau_2} M V$ follows by congruence of \sim_{τ_2} .

Similarly, we have $M' N' \sim_{\tau_2} M' V'$.

The conclusion (1) follows by transitivity of \sim_{τ_2} with (H).

Logical equivalence: application

Assume $not \triangleq \lambda x:B. \text{if } x \text{ then ff else tt}$

and $M \triangleq \lambda x:B. \lambda y:\tau. \lambda z:\tau. \text{if } not \ x \text{ then } y \text{ else } z$

and $M' \triangleq \lambda x:B. \lambda y:\tau. \lambda z:\tau. \text{if } x \text{ then } z \text{ else } y$

Show that $M \cong_{B \rightarrow \tau \rightarrow \tau \rightarrow \tau} M'$ (C).

It suffices to show $M \ V_0 \ V_1 \ V_2 \sim_{\tau} M' \ V'_0 \ V'_1 \ V'_2$ whenever $V_0 \sim_B V'_0$ and $V_1 \sim_{\tau} V'_1$ and $V_2 \sim_{\tau} V'_2$.

By inverse reduction, it suffices to show

$$\text{if } not \ V_0 \text{ then } V_1 \text{ else } V_2 \sim_{\tau} \text{if } V'_0 \text{ then } V'_2 \text{ else } V'_1$$

By cases on V_0 .

Case V_0 is tt: Then $not \ V_0 \longrightarrow^* \text{ff}$ and thus $M \longrightarrow^* V_2$ while $M' \longrightarrow^* V_2$. Then (C) follows by inverse reduction and $V_2 \sim_{\tau} V'_2$.

Case V_0 is ff: is symmetric.



Contents

- Introduction
- Normalization of stlc
- Observational equivalence in stlc
- Logical relations in stlc
- Logical relations in F

Observational equivalence

We now extend the notion of logical equivalence to System F.

$$\tau ::= \dots \mid \alpha \mid \forall \alpha. \tau \qquad M ::= \dots \mid \Lambda \alpha. M \mid M \tau$$

We write typing contexts $\Delta; \Gamma$ where Δ binds variables and Γ binds program variables.

Typing of contexts becomes $\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau')$.

Observational equivalence

We defined $\Delta; \Gamma \vdash M \cong M' : \tau$ as

$$\forall \mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\emptyset; \emptyset \triangleright B), \mathcal{C}[M] \simeq \mathcal{C}[M']$$

As before, write $M \cong_{\tau} N$ for $\emptyset, \emptyset \vdash M \cong N : \tau$ (in particular, τ is closed).

Logical equivalence

For closed terms (no free program variables)

- We need to give the semantics of polymorphic types $\forall\alpha. \tau$
- Problem: We cannot do it in terms of the semantics of instances $\tau[\alpha \mapsto \sigma]$ since the semantics is defined by induction on types.
- Solution: we give the semantics of terms with open types—in some suitable environment that interprets type variables by logical relations.

For simple types, we defined logical relations and observed that

- they respect observational equivalence
- they are closed by inverse reduction

We require that relations used to interpret type variables satisfy those properties.

Logical equivalence

Definition A relation R between closed expressions of closed types ρ and ρ' is admissible, and we write $R : \rho \leftrightarrow \rho'$, if:

- It respects observational equivalence: If $R(M, M')$ and $N \cong_{\rho} M$ and $N' \cong_{\rho'} M'$, then $R(N, N')$.
- It is closed under inverse reduction: If $R(M, M')$ and $N \longrightarrow^* M$ and $N' \longrightarrow^* M'$, then $R(N, N')$.

Given a sequence of type variables Δ , let δ and δ' be maps from $\text{dom}(\Delta)$ to closed types and let η be a map from $\text{dom}(\Delta)$ that sends each type variable α to an admissible relation between values of closed types $\delta(\alpha)$ and $\delta'(\alpha)$. We write $\eta : \delta \leftrightarrow_{\Delta} \delta'$ for such a relation.



Example of admissible relations

Take

$$\Delta \triangleq \alpha \qquad \delta \triangleq \alpha \mapsto \mathbf{B} \qquad \delta' \triangleq \alpha \mapsto \mathbb{Z}$$

Then $R: \mathit{bool} \leftrightarrow \mathit{int}$ may be the *closure by inverse reduction* of

$$\{(\mathit{tt}, 0)\} \cup \{(\mathit{ff}, n) \mid n \in \mathbb{Z}^*\}$$

where integers may be used to simulate booleans.

Allows to relate values at different types.



Logical equivalence for closed terms with open types

Assume $\eta : \delta \leftrightarrow_{\Delta} \delta'$ and $M : \delta(\tau)$ and $M' : \delta'(\tau)$.

We defined $M \sim_{\tau} M' [\eta : \delta \leftrightarrow \delta']$ by induction on τ as follows:

$$\begin{array}{ll}
 M \sim_{\mathbf{B}} M' [\eta : \delta \leftrightarrow \delta'] & \text{iff } M \simeq M' \\
 M \sim_{\tau_1 \rightarrow \tau_2} M' [\eta : \delta \leftrightarrow \delta'] & \text{iff for all } N \sim_{\tau_1} N' [\eta : \delta \leftrightarrow \delta'], \\
 & M N \sim_{\tau_2} M' N' [\eta : \delta \leftrightarrow \delta'] \\
 M \sim_{\alpha} M' [\eta : \delta \leftrightarrow \delta'] & \text{iff } \eta(\alpha)(M, M') \\
 M \sim_{\forall \alpha. \tau} M' [\eta : \delta \leftrightarrow \delta'] & \text{iff for all } \rho, \rho', R : \rho \leftrightarrow \rho', \\
 & M \rho \sim_{\tau} M' \rho' \\
 & [(\eta, \alpha \mapsto R) : (\delta, \alpha \mapsto \rho) \leftrightarrow (\delta', \alpha \mapsto \rho')]
 \end{array}$$

Logical equivalence for closed terms with open types

Assume $\eta : \delta \leftrightarrow_{\Delta} \delta'$ and $M : \delta(\tau)$ and $M' : \delta'(\tau)$.

We defined $M \sim_{\tau} M' [\eta : \delta \leftrightarrow \delta']$ by induction on τ as follows:

$$M \sim_{\mathbf{B}} M' [\eta] \quad \text{iff} \quad M \simeq M'$$

$$M \sim_{\tau_1 \rightarrow \tau_2} M' [\eta] \quad \text{iff} \quad \text{for all } N \sim_{\tau_1} N' [\eta], \quad M N \sim_{\tau_2} M' N' [\eta]$$

$$M \sim_{\alpha} M' [\eta] \quad \text{iff} \quad \eta(\alpha)(M, M')$$

$$M \sim_{\forall \alpha. \tau} M' [\eta] \quad \text{iff} \quad \text{for all } R : \rho \leftrightarrow \rho', \quad M \rho \sim_{\tau} M' \rho' [\eta, \alpha \mapsto R]$$

With implicit notations...



Logical equivalence for open terms

Definition If $\Delta; \Gamma \vdash M, M' : \tau$ we define $\Delta; \Gamma \vdash M \sim M' : \tau$ as

$$\forall \eta : \delta \leftrightarrow_{\Delta} \delta', \quad \forall \gamma \sim_{\Gamma} \gamma', \quad \gamma(\delta(M)) \sim_{\tau} \gamma'(\delta'(M')) \quad [\eta : \delta \leftrightarrow \delta']$$

(Notations are a bit heavy, but intuitions should remain simple.)

Notice We write $M \sim_{\tau} M'$ for $\emptyset; \emptyset \vdash M \sim M' : \tau$. In particular, τ is a closed type and M and M' are closed terms of type τ . By definition, this means $M \sim_{\tau} M' \quad [\emptyset : \emptyset \leftrightarrow \emptyset]$, which also coincide with the previous definition of logical relation for closed terms.

Properties

Closure under inverse reduction

If $M \sim_{\tau} M' [\eta : \delta \leftrightarrow \delta']$ and $N \longrightarrow^* M$ and $N' \longrightarrow^* M'$ (and $N : \delta(\tau)$), then $N \sim_{\tau} N' [\eta : \delta \leftrightarrow \delta']$.

Proof by induction on τ .

Similar to the monomorphic case, except for:

Case τ is $\forall\alpha.\sigma$:

To show $N \sim_{\tau} M' [\eta : \delta \leftrightarrow \delta']$,

we assume $R : \rho \leftrightarrow \rho'$ and show $N \rho \sim_{\sigma} M' \rho' [\eta, \alpha \mapsto R]$.

Since $N \rho \longrightarrow M \rho$,

by induction hypothesis it suffices to show $M \rho \sim_{\sigma} M' \rho' [\eta, \alpha \mapsto R]$, which follows from $M \sim_{\tau} M' [\eta : \delta \leftrightarrow \delta']$.

Properties

Respect for observational equivalence

If $M \sim_{\tau} M'$ $[\eta : \delta \leftrightarrow \delta']$ and $N \cong_{\delta(\tau)} M$ and $N' \cong_{\delta'(\tau)} M'$
 then $N \sim_{\tau} N'$ $[\eta : \delta \leftrightarrow \delta']$.

Proof by induction on τ .

Assume $M \sim_{\tau} M'$ $[\eta : \delta \leftrightarrow \delta']$ (1) and $N \cong_{\delta(\tau)} M$ (2). We show
 $N \sim_{\tau} M'$ $[\eta : \delta \leftrightarrow \delta']$.

Case τ is $\forall \alpha. \sigma$:

We assume $R : \rho \leftrightarrow \rho'$ and show $N \rho \sim_{\sigma} M' \rho' [\eta, \alpha \mapsto R]$.

Since $N \rho \cong_{\delta(\tau)} M \rho$ (by (2) as \cong is a congruence),

by induction hypothesis it suffices to show $M \rho \sim_{\sigma} M' \rho' [\eta, \alpha \mapsto R]$,
 which follows from (1).

Properties

Corollary The relation $M \sim_{\tau} M' [\eta : \delta \leftrightarrow \delta']$ is an admissible relation between expressions of closed types $\delta(\tau)$ and $\delta'(\tau)$.

(Useful, as we may take \sim_{τ} for the default relation.)

Properties

Lemma (respect for observational equivalence)

If $\Delta; \Gamma \vdash M \sim M' : \tau$ and $\Delta; \Gamma \vdash M \cong N : \tau$ and $\Delta; \Gamma \vdash M' \cong N' : \tau$, then $\Delta; \Gamma \vdash N \sim N' : \tau$

Lemma (Compositionality)

$M \sim_{\tau[\alpha \mapsto \sigma]} M' [\eta : \delta \leftrightarrow \delta']$ iff

$$M \sim_{\tau} M' [(\eta, \alpha \mapsto R) : (\delta, \alpha \mapsto \delta(\sigma)) \leftrightarrow (\delta', \alpha \mapsto \delta'(\sigma))]$$

where $R : \delta(s) \leftrightarrow \delta'(s)$ is defined by

$$R(N, N') \iff N \sim_{\sigma} N' [\eta : \delta \leftrightarrow \delta']$$

Proof by structural induction on τ .

Case τ is α :

Parametricity

Theorem (reflexivity) If $\Delta; \Gamma \vdash M : \tau$ then $\Delta; \Gamma \vdash M \sim M : \tau$.
(Also called parametricity or the fundamental theorem.)

Proof by induction on the typing derivation.

Properties

Theorem

Logical equivalence and observational equivalence coincide.

i.e. $\Delta; \Gamma \vdash M \sim M' : \tau$ iff $\Delta; \Gamma \vdash M \cong M' : \tau$.

As a particular case, $M \sim_{\tau} M'$ iff $M \cong_{\tau} M'$.

Properties

Extensionality

$M \cong_{\tau_1 \rightarrow \tau_2} M'$ iff for all $M_1 : \tau_1$, $M M_1 \cong_{\tau_2} M' M_1$.

$M \cong_{\forall \alpha. \tau} M'$ iff for all closed type ρ , $M \rho \cong_{\tau[\alpha \mapsto \rho]} M' \rho$.

Proof. Forward direction is immediate as \cong is a congruence.

Case Value abstraction: It suffices to show $M \sim_{\tau_1 \rightarrow \tau_2} M'$. That is, given $M_1 \sim_{\tau_1} M'_1$ (1), we show $M M_1 \sim_{\tau_2} M' M'_1$ (2). By assumption, we have $M M'_1 \cong_{\tau_2} M' M'_1$ (3). By the fundamental lemma, we have $M \sim_{\tau_1 \rightarrow \tau_2} M$. Hence, from (1), we get $M M_1 \sim_{\tau_2} M M'_1$.

We conclude (2) by respect for observational equivalence with (3).

Case Type abstraction: It suffices to show $M \sim_{\forall \alpha. \tau} M'$. That is, given $R : \rho \leftrightarrow \rho'$, we show $M \rho \sim_{\forall \alpha. \tau} M' \rho'$ (4). By assumption, we have $M \rho' \cong_{\tau_2} M' \rho'$ (5). By the fundamental lemma, we have $M \sim_{\forall \alpha. \tau} M$. Hence, we have $M \rho \sim_{\tau_2} M \rho'$. We conclude (4) by respect for observational equivalence with (5).

Properties

Identity extension

Let $\eta : \delta \leftrightarrow \delta$ where $\eta(\alpha)$ is observational equivalence at type $\delta(\alpha)$ for all $\alpha \in \text{dom}(\delta)$. Then $M \sim_{\tau} M' \ [\eta : \delta \leftrightarrow \delta]$ iff $M \cong_{\delta(\tau)} M'$.

Applications

Inhabitants of $\forall\alpha. \alpha \rightarrow \alpha$

Fact If $M : \forall\alpha. \alpha \rightarrow \alpha$, then $M \cong_{\forall\alpha. \alpha \rightarrow \alpha} id$ where $id \triangleq \Lambda\alpha. \lambda x:\alpha. x$.

Proof By extensionality, it suffices to show that for any ρ and $N : \rho$ we have $M \rho N \cong_{\rho} id \rho N$. In fact, by closure by inverse reduction, it suffices to show $M \rho N \cong_{\rho} N$ or, equivalently, $M \rho N \sim_{\rho} N$ (1).

By parametricity, we have $M \sim_{\forall\alpha. \alpha \rightarrow \alpha} M$.

The relation $R : \rho \leftrightarrow \rho$ defined as $R(P, P')$ iff $P \sim_{\rho} N \sim_{\rho} P'$ is admissible. Thus $M \rho \sim_{\alpha \rightarrow \alpha} M \rho [\eta]$ where η is $\alpha \mapsto R$.

By parametricity, we have $N \sim_{\rho} N$, hence $R(N, N)$, thus $N \sim_{\alpha} N [\eta]$.

Therefore $M \rho N \sim_{\alpha} M \rho N [\eta]$, i.e. $R(M \rho N, M \rho N)$, by definition of \sim_{α} , which implies (1) by definition of R .

Applications

Inhabitants of $\forall \alpha. \alpha \rightarrow \alpha \rightarrow \alpha$

Fact Let σ be $\forall \alpha. \alpha \rightarrow \alpha \rightarrow \alpha$ If $M : \sigma$, then either

$M \cong_{\sigma} M_1 \triangleq \Lambda \alpha. \lambda x_1 : \alpha. \lambda x_2 : \alpha. x_1$ or $M \cong_{\sigma} M_2 \triangleq \Lambda \alpha. \lambda x_1 : \alpha. \lambda x_2 : \alpha. x_2$

Proof By extensionality, it suffices to show for either $i = 1$ or $i = 2$ that for any closed type ρ and $N_1, N_2 : \rho$, we have $M \rho N_1 N_2 \cong_{\sigma} M_i \rho N_1 N_2$, or, by closure by inverse reduction and replacing observational by logical equivalence that $M \rho N_1 N_2 \cong_{\sigma} N_i$ (1).

?



Applications

Inhabitants of $\forall\alpha. \alpha \rightarrow \alpha \rightarrow \alpha$

Fact Let σ be $\forall\alpha. \alpha \rightarrow \alpha \rightarrow \alpha$ If $M : \sigma$, then either

$M \cong_{\sigma} M_1 \triangleq \Lambda\alpha. \lambda x_1 : \alpha. \lambda x_2 : \alpha. x_1$ or $M \cong_{\sigma} M_2 \triangleq \Lambda\alpha. \lambda x_1 : \alpha. \lambda x_2 : \alpha. x_2$

Proof By extensionality, it suffices to show for either $i = 1$ or $i = 2$ that for any closed type ρ and $N_1, N_2 : \rho$, we have $M \rho N_1 N_2 \cong_{\sigma} M_i \rho N_1 N_2$, or, by closure by inverse reduction and replacing observational by logical equivalence that $M \rho N_1 N_2 \cong_{\sigma} N_i$ (**1**).

Let ρ and $N_1, N_2 : \rho$ be fixed. The relation $R : B \leftrightarrow \rho$ defined as $R(P, P')$ iff $(P \sim_B \text{tt} \wedge P' \sim_{\rho} N_1)$ or $(P \sim_B \text{ff} \wedge P' \sim_{\rho} N_2)$ is admissible.

?



Applications

Inhabitants of $\forall\alpha. \alpha \rightarrow \alpha \rightarrow \alpha$

Fact Let σ be $\forall\alpha. \alpha \rightarrow \alpha \rightarrow \alpha$ If $M : \sigma$, then either

$M \cong_{\sigma} M_1 \triangleq \Lambda\alpha. \lambda x_1 : \alpha. \lambda x_2 : \alpha. x_1$ or $M \cong_{\sigma} M_2 \triangleq \Lambda\alpha. \lambda x_1 : \alpha. \lambda x_2 : \alpha. x_2$

Proof By extensionality, it suffices to show for either $i = 1$ or $i = 2$ that for any closed type ρ and $N_1, N_2 : \rho$, we have $M \rho N_1 N_2 \cong_{\sigma} M_i \rho N_1 N_2$, or, by closure by inverse reduction and replacing observational by logical equivalence that $M \rho N_1 N_2 \cong_{\sigma} N_i$ (**1**).

Let ρ and $N_1, N_2 : \rho$ be fixed. The relation $R : \mathbb{B} \leftrightarrow \rho$ defined as $R(P, P')$ iff $(P \sim_{\mathbb{B}} \text{tt} \wedge P' \sim_{\rho} N_1)$ or $(P \sim_{\mathbb{B}} \text{ff} \wedge P' \sim_{\rho} N_2)$ is admissible. Moreover, we have $\text{tt} \sim_{\sigma} N_1 [\eta]$ where η is $\alpha \mapsto R$, since $R(\text{tt}, N_1)$ and, similarly, $\text{ff} \sim_{\sigma} N_2 [\eta]$.

We have $M \sim_{\sigma} M$ by parametricity.

?



Applications

Inhabitants of $\forall\alpha. \alpha \rightarrow \alpha \rightarrow \alpha$

Fact Let σ be $\forall\alpha. \alpha \rightarrow \alpha \rightarrow \alpha$. If $M : \sigma$, then either

$M \cong_{\sigma} M_1 \triangleq \Lambda\alpha. \lambda x_1 : \alpha. \lambda x_2 : \alpha. x_1$ or $M \cong_{\sigma} M_2 \triangleq \Lambda\alpha. \lambda x_1 : \alpha. \lambda x_2 : \alpha. x_2$

Proof By extensionality, it suffices to show for either $i = 1$ or $i = 2$ that for any closed type ρ and $N_1, N_2 : \rho$, we have $M \rho N_1 N_2 \cong_{\sigma} M_i \rho N_1 N_2$, or, by closure by inverse reduction and replacing observational by logical equivalence that $M \rho N_1 N_2 \cong_{\sigma} N_i$ (1).

Let ρ and $N_1, N_2 : \rho$ be fixed. The relation $R : B \leftrightarrow \rho$ defined as $R(P, P')$ iff $(P \sim_B \text{tt} \wedge P' \sim_{\rho} N_1)$ or $(P \sim_B \text{ff} \wedge P' \sim_{\rho} N_2)$ is admissible. Moreover, we have $\text{tt} \sim_{\sigma} N_1 [\eta]$ where η is $\alpha \mapsto R$, since $R(\text{tt}, N_1)$ and, similarly, $\text{ff} \sim_{\sigma} N_2 [\eta]$.

We have $M \sim_{\sigma} M$ by parametricity. Hence, $M B \text{tt ff} \sim_{\alpha} M \rho N_1 N_2 [\eta]$, i.e. $R(M B \text{tt ff}, M \rho N_1 N_2)$, which means:

$$\vee \begin{cases} M B \text{tt ff} \sim_B \text{tt} \wedge M \rho N_1 N_2 \sim_{\rho} N_1 \\ M B \text{tt ff} \sim_B \text{ff} \wedge M \rho N_1 N_2 \sim_{\rho} N_2 \end{cases}$$

?



Applications

Inhabitants of $\forall \alpha. \alpha \rightarrow \alpha \rightarrow \alpha$

Fact Let σ be $\forall \alpha. \alpha \rightarrow \alpha \rightarrow \alpha$. If $M : \sigma$, then either

$M \cong_{\sigma} M_1 \triangleq \Lambda \alpha. \lambda x_1 : \alpha. \lambda x_2 : \alpha. x_1$ or $M \cong_{\sigma} M_2 \triangleq \Lambda \alpha. \lambda x_1 : \alpha. \lambda x_2 : \alpha. x_2$

Proof By extensionality, it suffices to show for either $i = 1$ or $i = 2$ that for any closed type ρ and $N_1, N_2 : \rho$, we have $M \rho N_1 N_2 \cong_{\sigma} M_i \rho N_1 N_2$, or, by closure by inverse reduction and replacing observational by logical equivalence that $M \rho N_1 N_2 \cong_{\sigma} N_i$ (1).

Let ρ and $N_1, N_2 : \rho$ be fixed. The relation $R : B \leftrightarrow \rho$ defined as $R(P, P')$ iff $(P \sim_B \text{tt} \wedge P' \sim_{\rho} N_1)$ or $(P \sim_B \text{ff} \wedge P' \sim_{\rho} N_2)$ is admissible. Moreover, we have $\text{tt} \sim_{\sigma} N_1 [\eta]$ where η is $\alpha \mapsto R$, since $R(\text{tt}, N_1)$ and, similarly, $\text{ff} \sim_{\sigma} N_2 [\eta]$.

We have $M \sim_{\sigma} M$ by parametricity. Hence, $M B \text{tt ff} \sim_{\alpha} M \rho N_1 N_2 [\eta]$, i.e. $R(M B \text{tt ff}, M \rho N_1 N_2)$, which means:

$$\bigvee \begin{cases} M B \text{tt ff} \sim_B \text{tt} \wedge M \rho N_1 N_2 \sim_{\rho} N_1 \\ M B \text{tt ff} \sim_B \text{ff} \wedge M \rho N_1 N_2 \sim_{\rho} N_2 \end{cases}$$

Since, $M B \text{tt ff}$ is independent of ρ , N_1 , and N_2 , this actually shows (1).

Applications

Inhabitants of $\forall \alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$

?

Applications

Inhabitants of $\forall\alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$

Fact Let σ be $\forall\alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$. If $M : \sigma$, then $M \cong_{\sigma} M_n$ for some integer n , where $M_n \triangleq \Lambda\alpha. \lambda f:\alpha \rightarrow \alpha. \lambda x:\alpha. (f \alpha)^n (x \alpha)$

?



Applications

Inhabitants of $\forall\alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$

Fact Let σ be $\forall\alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$. If $M : \sigma$, then $M \cong_{\sigma} M_n$ for some integer n , where $M_n \triangleq \Lambda\alpha. \lambda f:\alpha \rightarrow \alpha. \lambda x:\alpha. (f \ \alpha)^n \ (x \ \alpha)$

That is, the inhabitants of $\forall\alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$ are the Church naturals.



Applications

Inhabitants of $\forall\alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$

Fact Let σ be $\forall\alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$. If $M : \sigma$, then $M \cong_{\sigma} M_n$ for some integer n , where $M_n \triangleq \Lambda\alpha. \lambda f:\alpha \rightarrow \alpha. \lambda x:\alpha. (f \alpha)^n (x \alpha)$

Proof

?

Applications

Inhabitants of $\forall\alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$

Fact Let σ be $\forall\alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$. If $M : \sigma$, then $M \cong_{\sigma} M_n$ for some integer n , where $M_n \triangleq \Lambda\alpha. \lambda f:\alpha \rightarrow \alpha. \lambda x:\alpha. (f \alpha)^n (x \alpha)$

Proof By extensionality, it suffices to show that there exists n such for any closed type ρ and terms $N_1 : \rho \rightarrow \rho$ and $N_2 : \rho$, we have $M \rho N_1 N_2 \cong_{\rho} M_n \rho N_1 N_2$, or, by closure by inverse reduction and replacing observational by logical equivalence, $M \rho N_1 N_2 \sim_{\rho} M_n \rho N_1 N_2$, (1).

?



Applications

Inhabitants of $\forall\alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$

Fact Let σ be $\forall\alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$. If $M : \sigma$, then $M \cong_{\sigma} M_n$ for some integer n , where $M_n \triangleq \Lambda\alpha. \lambda f : \alpha \rightarrow \alpha. \lambda x : \alpha. (f \alpha)^n (x \alpha)$

Proof By extensionality, it suffices to show that there exists n such for any closed type ρ and terms $N_1 : \rho \rightarrow \rho$ and $N_2 : \rho$, we have $M \rho N_1 N_2 \cong_{\rho} M_n \rho N_1 N_2$, or, by closure by inverse reduction and replacing observational by logical equivalence, $M \rho N_1 N_2 \sim_{\rho} M_n \rho N_1 N_2$, (1).

Let ρ and $N_1, N_2 : \rho$ be fixed. Let Z and S be M_0 and M_1 .

The relation $R : \sigma \leftrightarrow \rho$ defined as $R(P, P')$ iff $\exists k, P \sim_{\sigma} S^k Z \wedge P' \sim_{\rho} N_1^k N_2$ is admissible. We have $Z \sim_{\alpha} N_2 [\eta]$ where η is $\alpha \mapsto R$, since $R(Z, N_2)$.

We also have $S \sim_{\alpha} N_1 [\eta]$. Indeed,

?

Applications

Inhabitants of $\forall\alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$

Fact Let σ be $\forall\alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$. If $M : \sigma$, then $M \cong_{\sigma} M_n$ for some integer n , where $M_n \triangleq \Lambda\alpha. \lambda f : \alpha \rightarrow \alpha. \lambda x : \alpha. (f \ \alpha)^n \ (x \ \alpha)$

Proof By extensionality, it suffices to show that there exists n such for any closed type ρ and terms $N_1 : \rho \rightarrow \rho$ and $N_2 : \rho$, we have $M \ \rho \ N_1 \ N_2 \cong_{\rho} M_n \ \rho \ N_1 \ N_2$, or, by closure by inverse reduction and replacing observational by logical equivalence, $M \ \rho \ N_1 \ N_2 \sim_{\rho} M_n \ \rho \ N_1 \ N_2$, (1).

Let ρ and $N_1, N_2 : \rho$ be fixed. Let Z and S be M_0 and M_1 .

The relation $R : \sigma \leftrightarrow \rho$ defined as $R(P, P')$ iff $\exists k, P \sim_{\sigma} S^k \ Z \ \wedge \ P' \sim_{\rho} N_1^k \ N_2$ is admissible. We have $Z \sim_{\alpha} N_2 \ [\eta]$ where η is $\alpha \mapsto R$, since $R(Z, N_2)$.

We also have $S \sim_{\alpha} N_1 \ [\eta]$. Indeed, assume $N \sim_{\alpha} N' \ [\eta]$, i.e. $R(N, N')$. There exists k such that (N, N') is $(S^k \ Z, N_1^k \ N_2)$. Then, $(S \ N, N_1 \ N')$ is $(S^{k+1} \ Z, N_1^{k+1} \ N_2)$. Therefore $S^{k+1} \ Z \sim_{\alpha} N_1 \ N' \ [\eta]$. (A key to the proof.)

?

Applications

Inhabitants of $\forall\alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$

Fact Let σ be $\forall\alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$. If $M : \sigma$, then $M \cong_{\sigma} M_n$ for some integer n , where $M_n \triangleq \Lambda\alpha. \lambda f : \alpha \rightarrow \alpha. \lambda x : \alpha. (f \alpha)^n (x \alpha)$

Proof By extensionality, it suffices to show that there exists n such for any closed type ρ and terms $N_1 : \rho \rightarrow \rho$ and $N_2 : \rho$, we have $M \rho N_1 N_2 \cong_{\rho} M_n \rho N_1 N_2$, or, by closure by inverse reduction and replacing observational by logical equivalence, $M \rho N_1 N_2 \sim_{\rho} M_n \rho N_1 N_2$, (1).

Let ρ and $N_1, N_2 : \rho$ be fixed. Let Z and S be M_0 and M_1 .

The relation $R : \sigma \leftrightarrow \rho$ defined as $R(P, P')$ iff $\exists k, P \sim_{\sigma} S^k Z \wedge P' \sim_{\rho} N_1^k N_2$ is admissible. We have $Z \sim_{\alpha} N_2 [\eta]$ where η is $\alpha \mapsto R$, since $R(Z, N_2)$.

We also have $S \sim_{\alpha} N_1 [\eta]$. Indeed, assume $N \sim_{\alpha} N' [\eta]$, i.e. $R(N, N')$. There exists k such that (N, N') is $(S^k Z, N_1^k N_2)$. Then, $(S N, N_1 N')$ is $(S^{k+1} Z, N_1^{k+1} N_2)$. Therefore $S^{k+1} Z \sim_{\alpha} N_1 N' [\eta]$. (A key to the proof.)

By parametricity, we have $M \sim_{\sigma} M$.

?

Applications

Inhabitants of $\forall\alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$

Fact Let σ be $\forall\alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$. If $M : \sigma$, then $M \cong_{\sigma} M_n$ for some integer n , where $M_n \triangleq \Lambda\alpha. \lambda f : \alpha \rightarrow \alpha. \lambda x : \alpha. (f \alpha)^n (x \alpha)$

Proof By extensionality, it suffices to show that there exists n such for any closed type ρ and terms $N_1 : \rho \rightarrow \rho$ and $N_2 : \rho$, we have $M \rho N_1 N_2 \cong_{\rho} M_n \rho N_1 N_2$, or, by closure by inverse reduction and replacing observational by logical equivalence, $M \rho N_1 N_2 \sim_{\rho} M_n \rho N_1 N_2$, (1).

Let ρ and $N_1, N_2 : \rho$ be fixed. Let Z and S be M_0 and M_1 .

The relation $R : \sigma \leftrightarrow \rho$ defined as $R(P, P') \text{ iff } \exists k, P \sim_{\sigma} S^k Z \wedge P' \sim_{\rho} N_1^k N_2$ is admissible. We have $Z \sim_{\alpha} N_2 [\eta]$ where η is $\alpha \mapsto R$, since $R(Z, N_2)$.

We also have $S \sim_{\alpha} N_1 [\eta]$. Indeed, assume $N \sim_{\alpha} N' [\eta]$, i.e. $R(N, N')$. There exists k such that (N, N') is $(S^k Z, N_1^k N_2)$. Then, $(S N, N_1 N')$ is $(S^{k+1} Z, N_1^{k+1} N_2)$. Therefore $S^{k+1} Z \sim_{\alpha} N_1 N' [\eta]$. (A key to the proof.)

By parametricity, we have $M \sim_{\sigma} M$. Hence, $M \sigma S Z \sim_{\alpha} M \rho N_1 N_2 [\eta]$, i.e. $R(M \sigma S Z, M \rho N_1 N_2)$ which means that there exists n such that $M \sigma S Z \sim_{\sigma} S^n Z$ and $M \rho N_1 N_2 \sim_{\rho} N_1^n N_2$. Since, $M \sigma S Z$ is independent of n , we in fact have (1).

Representation independence

A client of an existential type $\exists\alpha. \tau$ should not see the difference between two implementations N_1 and N_2 of $\exists\alpha. \tau$ with witness types σ_1 and σ_2 .

A client M has type $\forall\alpha. \tau \rightarrow \tau'$ with $\alpha \notin \text{fv}(\tau')$; it must use the argument parametrically, and the result is independent of the witness type.

Assume that σ_1 and σ_2 are two closed representation types and $R : \sigma_1 \leftrightarrow \sigma_2$ is an admissible relation between them.

Suppose that $N_1 : \tau[\alpha \mapsto \sigma_1]$ and $N_2 : \tau[\alpha \mapsto \sigma_2]$ are two equivalent implementations of the operations, *i.e.* such that $N_1 \sim_{\tau} N_2$ $[\eta : \delta_1 \leftrightarrow \delta_2]$ where $\eta : \alpha \mapsto R$ and $\delta_1 : \alpha \mapsto \sigma_1$ and $\delta_2 : \alpha \mapsto \sigma_2$.

A client M satisfies $M \sim_{\forall\alpha. \tau \rightarrow \tau'} M$ $[\eta : \delta \leftrightarrow \delta']$ and, in fact, $M \sim_{\forall\alpha. \tau \rightarrow \tau'} M$ since α does not appear free in τ' .

Thus $M \sigma_1 N_1 \cong_{\tau'} M \sigma_2 N_2$. That is, the behavior with the implementation N_1 with representation type σ_1 is indistinguishable from the behavior with implementation N_2 with representation type σ_2 .



Bibliography I

(Most titles have a clickable mark “▷” that links to online versions.)

Robert Harper. *Practical Foundations for Programming Languages*. Cambridge University Press, 2012.

J. Roger Hindley and Jonathan P. Seldin. *Introduction to Combinators and Lambda-Calculus*. Cambridge University Press, 1986.

Benjamin C. Pierce. *Types and Programming Languages*. MIT Press, 2002.

- ▷ Andrew M. Pitts. [Parametric polymorphism and operational equivalence](#). *Mathematical Structures in Computer Science*, 10:321–359, 2000.
- ▷ John C. Reynolds. [Types, abstraction and parametric polymorphism](#). In *Information Processing 83*, pages 513–523. Elsevier Science, 1983.
- ▷ W. W. Tait. [Intensional interpretations of functionals of finite type i](#). *The Journal of Symbolic Logic*, 32(2):pp. 198–212, 1967. ISSN 00224812.

Bibliography II

- ▶ Philip Wadler. [Theorems for free!](#) In *Conference on Functional Programming Languages and Computer Architecture (FPCA)*, pages 347–359, September 1989.