MPRI 2.4, Functional programming and type systems

Gabriel Scherer
reusing course material from Didier Rémy

Logical relations and parametricity

Contents

What are logical relations?

So far, most proofs involving terms have proceeded by induction on the structure of *terms* (or, equivalently, on *typing derivations*).

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 (or sets of expressions)
- They can be used to prove type safety and strong normalization

Binary relations

- Binary relations relate pairs of 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.

Parametricity? Inhabitants of polymorphic types

In the presence of polymorphism (and in the absence of effects), a type can reveal a lot of information about the terms that inhabit it.

What can do a term of type $\forall\alpha.\,\alpha \to \mathit{int}$?

?

Parametricity? Inhabitants of polymorphic types

In the presence of polymorphism (and in the absence of effects), a type can reveal a lot of information about the terms that inhabit it.

What can do a term of type $\forall\alpha.\,\alpha \to \mathit{int}$?

- the function cannot examine its argument

so ?

Parametricity? Inhabitants of polymorphic types

In the presence of polymorphism (and in the absence of effects), a type can reveal a lot of information about the terms that inhabit it.

What can do a term of type $\forall\alpha.\, \alpha \to \mathit{int}$?

- the function cannot examine its argument
- it always returns the same integer

for example ?

Parametricity? Inhabitants of polymorphic types

In the presence of polymorphism (and in the absence of effects), a type can reveal a lot of information about the terms that inhabit it.

What can do a term of type $\forall\alpha.\,\alpha \to$ *int* ?

- the function cannot examine its argument
- it always returns the same integer
- $\lambda x.\,n$,
  $\lambda x.\,(\lambda y.\,y)\;n$,
  $\lambda x.\,(\lambda y.\,n)\;x$.
  etc.

What do they all have in common ?

Parametricity? Inhabitants of polymorphic types

In the presence of polymorphism (and in the absence of effects), a type can reveal a lot of information about the terms that inhabit it.

What can do a term of type $\forall\alpha.\,\alpha \to$ *int* ?

- the function cannot examine its argument
- it always returns the same integer
- $\lambda x.\,n$,
  $\lambda x.\,(\lambda y.\,y)\;n$,
  $\lambda x.\,(\lambda y.\,n)\;x$.
  etc.
- they are all $\beta\eta$-equivalent to the term $\lambda x.\,n$

Parametricity? Inhabitants of polymorphic types

In the presence of polymorphism (and in the absence of effects), a type can reveal a lot of information about the terms that inhabit it.

A term of type $\forall\alpha.\,\alpha \to \mathit{int}$?

- ▷ behaves as $\lambda x.\,n$

Parametricity? Inhabitants of polymorphic types

In the presence of polymorphism (and in the absence of effects), a type can reveal a lot of information about the terms that inhabit it.

A term of type $\forall\alpha.\,\alpha \to \mathit{int}$?

- behaves as $\lambda x.\,n$

A term a of type $\forall\alpha.\,\alpha \to \alpha$?

?

Parametricity? Inhabitants of polymorphic types

In the presence of polymorphism (and in the absence of effects), a type can reveal a lot of information about the terms that inhabit it.

A term of type $\forall\alpha.\,\alpha \to \mathit{int}$?

- ▷ behaves as $\lambda x.\,n$

A term a of type $\forall\alpha.\,\alpha \to \alpha$?

- ▷ behaves as $\lambda x.\,x$

Parametricity? Inhabitants of polymorphic types

In the presence of polymorphism (and in the absence of effects), a type can reveal a lot of information about the terms that inhabit it.

A term of type $\forall \alpha. \alpha \to \mathit{int}$?

- behaves as $\lambda x. n$

A term a of type $\forall \alpha. \alpha \to \alpha$?

- behaves as $\lambda x. x$

A term type $\forall \alpha \beta. \alpha \to \beta \to \alpha$?

?

Parametricity? Inhabitants of polymorphic types

In the presence of polymorphism (and in the absence of effects), a type can reveal a lot of information about the terms that inhabit it.

A term of type $\forall\alpha.\, \alpha \to \mathit{int}$?

- ▷ behaves as $\lambda x.\, n$

A term a of type $\forall\alpha.\, \alpha \to \alpha$?

- ▷ behaves as $\lambda x.\, x$

A term type $\forall\alpha\beta.\, \alpha \to \beta \to \alpha$?

- ▷ behaves as $\lambda x.\, \lambda y.\, x$

A term type $\forall\alpha.\, \alpha \to \alpha \to \alpha$?

?

Parametricity? Inhabitants of polymorphic types

In the presence of polymorphism (and in the absence of effects), a type can reveal a lot of information about the terms that inhabit it.

A term of type $\forall\alpha.\,\alpha \to \mathit{int}$?

- behaves as $\lambda x.\, n$

A term a of type $\forall\alpha.\,\alpha \to \alpha$?

- behaves as $\lambda x.\, x$

A term type $\forall\alpha\beta.\,\alpha \to \beta \to \alpha$?

- behaves as $\lambda x.\, \lambda y.\, x$

A term type $\forall\alpha.\,\alpha \to \alpha \to \alpha$?

- behaves either as $\lambda x.\, \lambda y.\, x$ or $\lambda x.\, \lambda y.\, y$

Pametricity

Theorems for free

Similarly, the type of a polymorphic function may also reveal a *“free theorem”* about its behavior!

What properties may we learn from a function

$$\texttt{whoami} \;:\; \forall\alpha.\, \mathit{list}\,\alpha \to \mathit{list}\,\alpha$$

?

Pametricity

Theorems for free

Similarly, the type of a polymorphic function may also reveal a *"free theorem"* about its behavior!

What properties may we learn from a function

$$\texttt{whoami} : \forall\alpha.\, \mathit{list}\,\alpha \to \mathit{list}\,\alpha$$

▷ The length of the result depends only on the length of the argument

Pametricity

Theorems for free

Similarly, the type of a polymorphic function may also reveal a *“free theorem”* about its behavior!

What properties may we learn from a function

$$\texttt{whoami} : \forall\alpha.\, \mathit{list}\,\alpha \to \mathit{list}\,\alpha$$

- ▷ The length of the result depends only on the length of the argument
- ▷ All elements of the results are elements of the argument

Pametricity

Theorems for free

Similarly, the type of a polymorphic function may also reveal a *"free theorem"* about its behavior!

What properties may we learn from a function

$$\texttt{whoami} \; : \; \forall \alpha.\, \mathit{list}\, \alpha \to \mathit{list}\, \alpha$$

- ▷ The length of the result depends only on the length of the argument
- ▷ All elements of the results are elements of the argument
- ▷ The choice (i, j) of pairs such that i-th element of the result is the j-th element of the argument does not depend on the element itself.

Pametricity <span>Theorems for free</span>

Similarly, the type of a polymorphic function may also reveal a *"free theorem"* about its behavior!

What properties may we learn from a function

$$\texttt{whoami} \; : \; \forall \alpha.\, \mathit{list}\, \alpha \to \mathit{list}\, \alpha$$

- ▷ The length of the result depends only on the length of the argument
- ▷ All elements of the results are elements of the argument
- ▷ The choice (i, j) of pairs such that i-th element of the result is the j-th element of the argument does not depend on the element itself.
- ▷ the function is preserved by a transformation of its argument that preserves the shape of the argument

$$\forall f, x, \quad \texttt{whoami}\ (map\ f\ x) = map\ f\ (\texttt{whoami}\ x)$$

Pametricity Theorems for free

Similarly, the type of a polymorphic function may also reveal a *“free theorem”* about its behavior!

What properties may we learn from a function

$$\texttt{whoami} \; : \; \forall \alpha.\, \mathit{list}\,\alpha \to \mathit{list}\,\alpha$$

What property may we learn for the list sorting function?

$$\mathit{sort} \; : \; \forall \alpha.\, (\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\,\alpha \to \mathit{list}\,\alpha$$

?

Pametricity

Theorems for free

Similarly, the type of a polymorphic function may also reveal a *"free theorem"* about its behavior!

What properties may we learn from a function

$$\texttt{whoami} \; : \; \forall\alpha.\, \mathit{list}\,\alpha \to \mathit{list}\,\alpha$$

What property may we learn for the list sorting function?

$$\mathit{sort} \; : \; \forall\alpha.\,(\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\,\alpha \to \mathit{list}\,\alpha$$

If f is order-preserving, then sorting commutes with $map\ f$

$$(\forall x, y,\ \ \mathit{cmp}\,(f\ x)\,(f\ y) = \mathit{cmp}\ \ x\ y) \implies$$
$$\forall \ell,\ \ \mathit{sort}\ \ \mathit{cmp}\,(\mathit{map}\ f\ \ell) = \mathit{map}\ f\ (\mathit{sort}\ \mathit{cmp}\ \ell)$$

Pametricity

Theorems for free

Similarly, the type of a polymorphic function may also reveal a *“free theorem”* about its behavior!

What properties may we learn from a function

$$\texttt{whoami} \;:\; \forall\alpha.\, \mathit{list}\,\alpha \to \mathit{list}\,\alpha$$

What property may we learn for the list sorting function?

$$\mathit{sort} \;:\; \forall\alpha.\,(\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\,\alpha \to \mathit{list}\,\alpha$$

If f is order-preserving, then sorting commutes with $map\ f$

$$(\forall x, y,\;\; \mathit{cmp}_2\ (f\ x)\ (f\ y) = \mathit{cmp}_1\ \ x\ y) \implies \\ \forall \ell,\;\; \mathit{sort}\ \ \mathit{cmp}_2\ (\mathit{map}\ f\ \ell) = \mathit{map}\ f\ (\mathit{sort}\ \mathit{cmp}_1\ \ell)$$

Pametricity

Theorems for free

Similarly, the type of a polymorphic function may also reveal a *"free theorem"* about its behavior!

What properties may we learn from a function

$$\texttt{whoami} \;:\; \forall\alpha.\,\mathit{list}\,\alpha \to \mathit{list}\,\alpha$$

What property may we learn for the list sorting function?

$$\mathit{sort} \;:\; \forall\alpha.\,(\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\,\alpha \to \mathit{list}\,\alpha$$

If f is order-preserving, then sorting commutes with $map\ f$

$$(\forall x, y,\ \ \mathit{cmp}_2\ (f\ x)\ (f\ y) = \mathit{cmp}_1\ \ x\ y) \implies \\ \forall \ell,\ \ \mathit{sort}\ \ \mathit{cmp}_2\ (\mathit{map}\ f\ \ell) = \mathit{map}\ f\ (\mathit{sort}\ \mathit{cmp}_1\ \ell)$$

Application:

- If *sort* is correct on lists of integers, then it is correct on any list
- May be useful to reduce testing.

Pametricity <span>Theorems for free</span>

Similarly, the type of a polymorphic function may also reveal a *“free theorem”* about its behavior!

What properties may we learn from a function

$$\texttt{whoami} : \forall\alpha.\,\mathit{list}\,\alpha \to \mathit{list}\,\alpha$$

What property may we learn for the list sorting function?

$$\mathit{sort} : \forall\alpha.\,(\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\,\alpha \to \mathit{list}\,\alpha$$

If f is order-preserving, then sorting commutes with $map\ f$

$$(\forall x, y,\ \mathit{cmp}_2\ (f\ x)\ (f\ y) = \mathit{cmp}_1\ x\ y) \implies \forall \ell,\ \mathit{sort}\ \mathit{cmp}_2\ (\mathit{map}\ f\ \ell) = \mathit{map}\ f\ (\mathit{sort}\ \mathit{cmp}_1\ \ell)$$

Note that there are many other inhabitants of this type, but they all satisfy this free theorem.

Can you give a few?

Pametricity

Theorems for free

Similarly, the type of a polymorphic function may also reveal a *"free theorem"* about its behavior!

What properties may we learn from a function

$$\texttt{whoami} \;:\; \forall \alpha.\, \mathit{list}\,\alpha \to \mathit{list}\,\alpha$$

What property may we learn for the list sorting function?

$$\mathit{sort} \;:\; \forall \alpha.\, (\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\,\alpha \to \mathit{list}\,\alpha$$

If f is order-preserving, then sorting commutes with $map\ f$

$$(\forall x, y,\;\; \mathit{cmp}_2\ (f\ x)\ (f\ y) = \mathit{cmp}_1\ \ x\ y) \implies \\ \forall \ell,\;\; \mathit{sort}\ \ \mathit{cmp}_2\ (\mathit{map}\ f\ \ell) = \mathit{map}\ f\ (\mathit{sort}\ \mathit{cmp}_1\ \ell)$$

Note that there are many other inhabitants of this type, but they all satisfy this free theorem. (e.g., a function that sorts in reverse order, or a function that removes (or adds) duplicates).

Parametricity

This phenomenon was studied by Reynolds [1983] and by Wadler [1989; 2007], among others. Wadler's paper contains the 'free theorem' about the list sorting function.

An account based on an operational semantics is offered by Pitts [2000].

Bernardy et al. [2010] generalize the idea of testing polymorphic functions to arbitrary polymorphic types and show how testing any function can be restricted to testing it on (possibly infinitely many) particular values at some particular types.

Contents

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 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 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 λ_{st} 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 (ground) type τ by induction on types as follows:

$$\begin{aligned}
\mathcal{V}[\![\mathsf{B}]\!] &\triangleq \{\mathsf{tt}, \mathsf{ff}\} \\
\mathcal{V}[\![\tau_1 \to \tau_2]\!] &\triangleq \{\lambda x{:}\tau_1.\, M \mid \lambda x{:}\tau_1.\, M : \tau_1 \to \tau_2 \\
&\qquad\qquad \wedge\ \forall V \in \mathcal{V}[\![\tau_1]\!],\ (\lambda x{:}\tau_1.\, M)\, V \in \mathcal{E}[\![\tau_2]\!]\} \\
\mathcal{E}[\![\tau]\!] &\triangleq \{M \mid M : \tau \wedge \exists V \in \mathcal{V}[\![\tau]\!], M \Downarrow V\}
\end{aligned}$$

We write $M \Downarrow N$ for $M \longrightarrow^* N$.
The goal is to show that any closed expression of type τ is in $\mathcal{E}[\![\tau]\!]$.

Remarks

Although usual with logical relations, well-typedness is actually not required here and omitted: otherwise, we would have to carry unnecessary type-preservation proof obligations.

Calculus

Take the call-by-value λ_{st} 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 (ground) type τ by induction on types as follows:

$$\begin{aligned}\mathcal{V}[\![\mathsf{B}]\!] &\triangleq \{\mathsf{tt}, \mathsf{ff}\}\\ \mathcal{V}[\![\tau_1 \to \tau_2]\!] &\triangleq \{\lambda x{:}\tau_1.\, M \mid \forall V \in \mathcal{V}[\![\tau_1]\!],\ (\lambda x{:}\tau_1.\, M)\, V \in \mathcal{E}[\![\tau_2]\!]\}\\ \mathcal{E}[\![\tau]\!] &\triangleq \{M \mid \exists V \in \mathcal{V}[\![\tau]\!], M \Downarrow V\}\end{aligned}$$

We write $M \Downarrow N$ for $M \longrightarrow^* N$.
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 $M \Downarrow N$ 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 premise is 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 $\varnothing \vDash M : \tau$ means $M \in \mathcal{E}[\![\tau]\!]$.

We interpret free term variables of type τ as *closed values* in $\mathcal{V}[\![\tau]\!]$.

We interpret environments Γ as *closing substitutions* γ, *i.e.* mappings from term variables to *closed values*:

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

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

Fundamental Lemma

Theorem (fundamental lemma)
If $\Gamma \vdash M : \tau$ then $\Gamma \vDash M : \tau$.

Corollary (termination of well-typed terms):
If $\varnothing \vdash M : \tau$ then $M \in \mathcal{E}[\![\tau]\!]$.

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

Proof by induction on the typing derivation

Routine cases

Case $\Gamma \vdash \mathsf{tt} : \mathsf{B}$ *or* $\Gamma \vdash \mathsf{ff} : \mathsf{B}$: by definition, $\mathsf{tt}, \mathsf{ff} \in \mathcal{V}[\![\mathsf{B}]\!]$ and $\mathcal{V}[\![\mathsf{B}]\!] \subseteq \mathcal{E}[\![\mathsf{B}]\!]$.

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, $\Gamma \vdash M_1 : \tau_2 \to \tau$ 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 $\Gamma \vDash M_1 : \tau_2 \to \tau$ and $\Gamma \vDash M_2 : \tau_2$.
Thus $\gamma M_1 \in \mathcal{E}[\![\tau_2 \to \tau]\!]$ $(\mathbf{1})$ and $\gamma M_2 \in \mathcal{E}[\![\tau_2]\!]$ $(\mathbf{2})$.

By (2), there exists $V \in \mathcal{V}[\![\tau_2]\!]$ such that $\gamma M_2 \Downarrow V$.
Thus $(\gamma M_1)\ (\gamma M_2) \Downarrow (\gamma M_1)\ V \in \mathcal{E}[\![\tau]\!]$ by (1).

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

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

Proof by induction on the typing derivation (key case)

The interesting case

Case $\Gamma \vdash \lambda x{:}\tau_1.\, M : \tau_1 \to \tau$:

Assume $\gamma \in \mathcal{G}[\![\Gamma]\!]$.
We must show that $\gamma(\lambda x{:}\tau_1.\, M) \in \mathcal{E}[\![\tau_1 \to \tau]\!]$ (1)

That is, $\lambda x{:}\tau_1.\, \gamma M \in \mathcal{V}[\![\tau_1 \to \tau]\!]$ (we may assume $x \notin \mathrm{dom}(\gamma)$ *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 $\Gamma, x : \tau_1 \vDash M : \tau$ by IH on M. 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:

$$\mathcal{E}[\![\tau]\!] = \left\{M : \tau \mid \forall N, M \Downarrow N \implies \left(N \in \mathcal{V}[\![\tau]\!] \ \vee\ \exists N', N \longrightarrow N'\right)\right\}$$

Compare with

$$\mathcal{E}[\![\tau]\!] = \{M : \tau \mid \exists V \in \mathcal{V}[\![\tau]\!], M \Downarrow V\}$$

Exercise

Show type soundness with this semantics.

Contents

(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].
- Course notes taken by Lau Skorstengaard on Amal Ahmed's OPLSS lectures.

We assume a call-by-value operational semantics instead of call-by-name in [Harper, 2012].

When are two programs equivalent

?

When are two programs equivalent

$M \Downarrow N$?

?

When are two programs equivalent

$M \Downarrow N$?

$M \Downarrow V$ and $N \Downarrow V$?

?

When are two programs equivalent

$M \Downarrow N$?

$M \Downarrow V$ and $N \Downarrow V$?

But what if M and N are functions?

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

Idea

?

When are two programs equivalent

$M \Downarrow N$?

$M \Downarrow V$ and $N \Downarrow V$?

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 $M : \mathsf{B}$ and $N : \mathsf{B}$, then $M \simeq N$ iff there exists V such that $M \Downarrow V$ and $N \Downarrow V$. (Call $M \simeq N$ *behavioral equivalence*.)

To compare programs at other types, we

?

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 $M : \mathsf{B}$ and $N : \mathsf{B}$, then $M \simeq N$ iff there exists V such that $M \Downarrow V$ and $N \Downarrow V$. (Call $M \simeq N$ *behavioral equivalence*.)

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

Definition (observational equivalence)

$$\Gamma \vdash M \cong N : \tau \;\triangleq\; \forall \mathcal{C} : (\Gamma \rhd \tau) \rightsquigarrow (\varnothing \rhd \mathsf{B}),\ \mathcal{C}[M] \simeq \mathcal{C}[N]$$

Typing of contexts

$$\mathcal{C} : (\Gamma \rhd \tau) \rightsquigarrow (\Delta \rhd \sigma) \iff (\forall M,\ \Gamma \vdash M : \tau \implies \Delta \vdash \mathcal{C}[M] : \sigma)$$

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

We write $M \cong_\tau N$ for $\varnothing \vdash M \cong N : \tau$

Observational equivalence

Observational equivalence is the coarsiest consistent congruence, where:

- $\equiv$ is consistent if $\varnothing \vdash M \equiv N : \mathsf{B}$ implies $M \simeq N$.
- $\equiv$ is a congruence if it is an equivalence and is closed by context, *i.e.*

$$\Gamma \vdash M \equiv N : \tau \;\wedge\; \mathcal{C} : (\Gamma \rhd \tau) \rightsquigarrow (\Delta \rhd \sigma) \quad \Longrightarrow \quad \Delta \vdash \mathcal{C}[M] \equiv \mathcal{C}[N] : \sigma$$

Consistent: by definition, using the empty context.

Congruence: by compositionality of contexts.

Coarsiest: Assume $\equiv$ is a consistent congruence.

We assume $\Gamma \vdash M \equiv N : \tau$ (1) and show $\Gamma \vdash M \cong N : \tau$.

Let $\mathcal{C} : (\Gamma \rhd \tau) \rightsquigarrow (\varnothing \rhd \mathsf{B})$ (2). We must show that $\mathcal{C}[M] \simeq \mathcal{C}[N]$.
This follows by consistency applied to $\Gamma \vdash \mathcal{C}[M] \equiv \mathcal{C}[N] : \mathsf{B}$
which itself follows by congruence from (1) and (2).

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.

Contents

Logical equivalence for closed terms

Unary logical relations interpret types by predicates on (*i.e.* sets of) closed values of that type.

Binary relations interpret types by binary relations on closed values of that type, *i.e.* sets of pairs of related values of that type.

That is $\mathcal{V}[\![\tau]\!] \subseteq \mathsf{Val}(\tau) \times \mathsf{Val}(\tau)$.

Then, $\mathcal{E}[\![\tau]\!]$ is the closure of $\mathcal{V}[\![\tau]\!]$ by inverse reduction

We have $\mathcal{V}[\![\tau]\!] \subseteq \mathcal{E}[\![\tau]\!] \subseteq \mathsf{Exp}(\tau) \times \mathsf{Exp}(\tau)$.

Logical equivalence for closed terms

We recursively define two relations $\mathcal{V}[\![\tau]\!]$ and $\mathcal{E}[\![\tau]\!]$ between values of type τ and expressions of type τ by

$$\begin{aligned}
\mathcal{V}[\![\mathsf{B}]\!] &\triangleq \{(\mathsf{tt},\mathsf{tt}),(\mathsf{ff},\mathsf{ff})\}\\
\mathcal{V}[\![\tau \to \sigma]\!] &\triangleq \{(V_1,V_2) \mid V_1,V_2 : \tau \to \sigma \wedge \\
&\qquad \forall (W_1,W_2) \in \mathcal{V}[\![\tau]\!],\ (V_1\,W_1, V_2\,W_2) \in \mathcal{E}[\![\sigma]\!]\}\\
\mathcal{E}[\![\tau]\!] &\triangleq \{(M_1,M_2) \mid M_1,M_2 : \tau \wedge \\
&\qquad \exists (V_1,V_2) \in \mathcal{V}[\![\tau]\!],\ M_1 \Downarrow V_1 \wedge M_2 \Downarrow V_2\}
\end{aligned}$$

In the following we will leave the typing constraint in gray implicit (as a global condition for sets $\mathcal{V}[\![\cdot]\!]$ and $\mathcal{E}[\![\cdot]\!]$).

We also write

$M_1 \sim_\tau M_2$ for $(M_1,M_2) \in \mathcal{E}[\![\tau]\!]$ and
$V_1 \approx_\tau V_2$ for $(V_1,V_2) \in \mathcal{V}[\![\tau]\!]$.

Logical equivalence for closed terms

We recursively define two relations $\mathcal{V}[\![\tau]\!]$ and $\mathcal{E}[\![\tau]\!]$ between values of type τ and expressions of type τ by

$$\begin{aligned}
\mathcal{V}[\![\mathsf{B}]\!] &\triangleq \{(\mathsf{tt},\mathsf{tt}),(\mathsf{ff},\mathsf{ff})\}\\
\mathcal{V}[\![\tau\to\sigma]\!] &\triangleq \{(V_1,V_2) \mid V_1,V_2:\tau\to\sigma\wedge\\
&\qquad \forall(W_1,W_2)\in\mathcal{V}[\![\tau]\!],\ (V_1\,W_1,V_2\,W_2)\in\mathcal{E}[\![\sigma]\!]\}\\
\mathcal{E}[\![\tau]\!] &\triangleq \{(M_1,M_2) \mid M_1,M_2:\tau\wedge\\
&\qquad \Downarrow(M_1,M_2)\in\mathcal{V}[\![\tau]\!]\}
\end{aligned}$$

where $\Downarrow(M_1,M_2)$ *means* $\{(V_1,V_2) \mid M_i \Downarrow V_i\}$

In the following we will leave the typing constraint in gray implicit (as a global condition for sets $\mathcal{V}[\![\cdot]\!]$ and $\mathcal{E}[\![\cdot]\!]$).

We also write

$M_1 \sim_\tau M_2$ for $(M_1,M_2)\in\mathcal{E}[\![\tau]\!]$ and
$V_1 \approx_\tau V_2$ for $(V_1,V_2)\in\mathcal{V}[\![\tau]\!]$.

Logical equivalence for closed terms (variant)

In a language with non-termination

We change the definition of $\mathcal{E}[\![\tau]\!]$ to

$$\begin{aligned}\mathcal{E}[\![\tau]\!] \;\triangleq\; \Big\{(M_1, M_2) \;\Big|\; & M_1, M_2 : \tau \wedge \\ & \big(\forall V_1,\; M_1 \Downarrow V_1 \implies \exists V_2,\; M_2 \Downarrow V_2 \wedge (V_1, V_2) \in \mathcal{V}[\![\tau]\!]\big) \\ \wedge\, & \big(\forall V_2,\; M_2 \Downarrow V_2 \implies \exists V_1,\; M_1 \Downarrow V_1 \wedge (V_1, V_2) \in \mathcal{V}[\![\tau]\!]\big)\Big\}\end{aligned}$$

Notice

$$\begin{aligned}\mathcal{V}[\![\tau \to \sigma]\!] \;&\triangleq\; \{(V_1, V_2) \mid V_1, V_2 \vdash \tau \to \sigma \wedge \\ &\qquad \forall (W_1, W_2) \in \mathcal{V}[\![\tau]\!],\; (V_1\, W_1, V_2\, W_2) \in \mathcal{E}[\![\sigma]\!]\} \\ &= \{((\lambda x{:}\tau.\, M_1), (\lambda x{:}\tau.\, M_2)) \mid (\lambda x{:}\tau.\, M_1), (\lambda x{:}\tau.\, M_2) \vdash \tau \to \sigma \wedge \\ &\qquad \forall (W_1, W_2) \in \mathcal{V}[\![\tau]\!],\; ((\lambda x{:}\tau.\, M_1)\, W_1, (\lambda x{:}\tau.\, M_2)\, W_2) \in \mathcal{E}[\![\sigma]\!]\}\end{aligned}$$

Properties of logical equivalence for closed terms

Closure by reduction

By definition, since reduction is deterministic: Assume $M_1 \Downarrow N_1$ and $M_2 \Downarrow N_2$ and $(M_1, M_2) \in \mathcal{E}[\![\tau]\!]$, *i.e.* there exists $(V_1, V_2) \in \mathcal{V}[\![\tau]\!]$ $(\mathbf{1})$ such that $M_i \Downarrow V_i$. Since reduction is deterministic, we must have $M_i \Downarrow N_i \Downarrow V_i$. This, together with (1), implies $(N_1, M_2) \in \mathcal{E}[\![\tau]\!]$.

Closure by inverse reduction

Immediate, by construction of $\mathcal{E}[\![\tau]\!]$.

Corollaries

- If $(M_1, M_2) \in \mathcal{E}[\![\tau \to \sigma]\!]$ and $(N_1, N_2) \in \mathcal{E}[\![\tau]\!]$, then $(M_1\ N_1, M_2\ N_2) \in \mathcal{E}[\![\sigma]\!]$.
- To prove $(M_1, M_2) \in \mathcal{E}[\![\tau \to \sigma]\!]$, it suffices to show $(M_1\ V_1, M_2\ V_2) \in \mathcal{E}[\![\sigma]\!]$ for all $(V_1, V_2) \in \mathcal{V}[\![\tau]\!]$.

Properties of logical equivalence for closed terms

Consistency $(\sim_{\mathsf{B}}) \subseteq (\simeq)$

Immediate, by definition of $\mathcal{E}[\![\mathsf{B}]\!]$ and $\mathcal{V}[\![\mathsf{B}]\!] \subseteq (\simeq)$.

Lemma

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

Note: Reflexivity is not at all obvious.

Proof

We show it simultaneously for $\sim_\tau$ and $\approx_\tau$ by induction on type τ.

Properties of logical equivalence for closed terms (proof)

For $\sim_\tau$, the proof is immediate by transitivity and symmetry of $\approx_\tau$.

For $\approx_\tau$, it goes as follows.

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

Case τ is $\tau \to \sigma$:

By IH, symmetry and transitivity hold at types τ and σ.

For symmetry, assume $V_1 \approx_{\tau\to\sigma} V_2$ (H), we must show $V_2 \approx_{\tau\to\sigma} V_1$.

Assume $W_1 \approx_\tau W_2$. We must show $V_2\, W_1 \sim_\sigma V_1\, W_2$ (C). We have $W_2 \approx_\tau W_1$ by symmetry at type τ. By (H), we have $V_2\, W_2 \sim_\sigma V_1\, W_1$ and (C) follows by symmetry of $\sim$ at type σ.

For transitivity, assume $V_1 \approx_{\tau\to\sigma} V_2$ (H1) and $V_2 \approx_{\tau\to\sigma} V_3$ (H2). To show $V_1 \approx_{\tau\to\sigma} V_3$, we assume $W_1 \approx_\tau W_3$ and show $V_1\, W_1 \sim_\sigma V_3\, W_3$ (C).
By (H1), we have $V_1\, W_1 \sim_\sigma V_2\, W_3$ (C1).
By symmetry and transitivity of $\approx_\tau$ (IH), we get $W_3 \approx_\tau W_3$. *It's not reflexivity!*
By (H2), we have $V_2\, W_3 \sim_\sigma V_3\, W_3$ (C2).
(C) follows by transitivity of $\sim_\sigma$ applied to (C1) and (C2).

Logical equivalence for open terms

When $\Gamma \vdash M_1 : \tau$ and $\Gamma \vdash M_2 : \tau$, we wish to define a judgment $\Gamma \vdash M_1 \sim M_2 : \tau$ to mean that the open terms M_1 and M_2 are equivalent at type τ.

The solution is to interpret program variables of $\mathrm{dom}(\Gamma)$ by pairs of related values and typing contexts Γ by a set of (closing) bisubstitutions γ mapping variable type assignments to pairs of related values.

$$\begin{aligned} \mathcal{G}[\![\varnothing]\!] &\triangleq \{\varnothing\} \\ \mathcal{G}[\![\Gamma, x : \tau]\!] &\triangleq \{\gamma, x \mapsto (V_1, V_2) \mid \gamma \in \mathcal{G}[\![\Gamma]\!] \wedge (V_1, V_2) \in \mathcal{V}[\![\tau]\!]\} \end{aligned}$$

Given a bisubstitution γ, we write γ_i for the substitution that maps x to V_i whenever γ maps x to (V_1, V_2).

Definition

$$\Gamma \vdash M_1 \sim M_2 : \tau \iff \forall \gamma \in \mathcal{G}[\![\Gamma]\!],\ (\gamma_1 M_1, \gamma_2 M_2) \in \mathcal{E}[\![\tau]\!]$$

We also write $\vdash M_1 \sim M_2 : \tau$ or $M_1 \sim_\tau M_2$ for $\varnothing \vdash M_1 \sim M_2 : \tau$.

Properties of logical equivalence for open terms

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

Theorem (Reflexivity) *(also called the fundamental lemma))*

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

Proof By induction on the typing derivation, using compatibility lemmas.

Compatibility lemmas

C-TRUE
$$\Gamma \vdash \mathsf{tt} : \mathit{bool}$$

C-FALSE
$$\Gamma \vdash \mathsf{ff} : \mathit{bool}$$

C-VAR
$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau}$$

C-ABS
$$\frac{\Gamma, x : \tau \vdash M_1 : \sigma}{\Gamma \vdash \lambda x{:}\tau.\, M_1 : \tau \to \sigma}$$

C-APP
$$\frac{\Gamma \vdash M_1 : \tau \to \sigma \qquad \Gamma \vdash N_1 : \tau}{\Gamma \vdash M_1\ N_1 : \sigma}$$

C-IF
$$\frac{\Gamma \vdash M_1 : \mathsf{B} \qquad \Gamma \vdash N_1 : \tau \qquad \Gamma \vdash N_1' : \tau}{\Gamma \vdash \text{if } M_1 \text{ then } N_1 \text{ else } N_1' : \tau}$$

Fundamental lemma of logical equivalence

Theorem (Reflexivity) *(also called the fundamental lemma))*

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

Proof By induction on the typing derivation, using compatibility lemmas.

Compatibility lemmas

C-True
$$\Gamma \vdash \mathsf{tt} : \mathit{bool}$$

C-False
$$\Gamma \vdash \mathsf{ff} : \mathit{bool}$$

C-Var
$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau}$$

C-Abs
$$\frac{\Gamma, x : \tau \vdash M_1 : \sigma}{\Gamma \vdash \lambda x{:}\tau.\, M_1 : \tau \to \sigma}$$

C-App
$$\frac{\Gamma \vdash M_1 : \tau \to \sigma \qquad \Gamma \vdash N_1 : \tau}{\Gamma \vdash M_1\, N_1 : \sigma}$$

C-If
$$\frac{\Gamma \vdash M_1 : \mathsf{B} \qquad \Gamma \vdash N_1 : \tau \qquad \Gamma \vdash N_1' : \tau}{\Gamma \vdash \mathsf{if}\ M_1\ \mathsf{then}\ N_1\ \mathsf{else}\ N_1' : \tau}$$

Fundamental lemma of logical equivalence

Theorem (Reflexivity) *(also called the fundamental lemma))*

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

Proof By induction on the typing derivation, using compatibility lemmas.

Compatibility lemmas

C-True
$$\Gamma \vdash \mathsf{tt} \quad : \mathit{bool}$$

C-False
$$\Gamma \vdash \mathsf{ff} \quad : \mathit{bool}$$

C-Var
$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x \quad : \tau}$$

C-Abs
$$\frac{\Gamma, x : \tau \vdash M_1 \quad : \sigma}{\Gamma \vdash \lambda x{:}\tau.\, M_1 \quad : \tau \to \sigma}$$

C-App
$$\frac{\Gamma \vdash M_1 \quad : \tau \to \sigma \qquad \Gamma \vdash N_1 \quad : \tau}{\Gamma \vdash M_1\, N_1 \quad : \sigma}$$

C-If
$$\frac{\Gamma \vdash M_1 \quad : \mathsf{B} \qquad \Gamma \vdash N_1 \quad : \tau \qquad \Gamma \vdash N_1' \quad : \tau}{\Gamma \vdash \mathsf{if}\ M_1\ \mathsf{then}\ N_1\ \mathsf{else}\ N_1' \quad : \tau}$$

Fundamental lemma of logical equivalence

Theorem (Reflexivity) *(also called the fundamental lemma))*

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

Proof By induction on the typing derivation, using compatibility lemmas.

Compatibility lemmas

C-True
$$\Gamma \vdash \mathsf{tt} \sim \mathsf{tt} : \mathit{bool}$$

C-False
$$\Gamma \vdash \mathsf{ff} \sim \mathsf{ff} : \mathit{bool}$$

C-Var
$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x \sim x : \tau}$$

C-Abs
$$\frac{\Gamma, x : \tau \vdash M_1 \sim M_2 : \sigma}{\Gamma \vdash \lambda x{:}\tau.\, M_1 \sim \lambda x{:}\tau.\, M_2 : \tau \to \sigma}$$

C-App
$$\frac{\Gamma \vdash M_1 \sim M_2 : \tau \to \sigma \qquad \Gamma \vdash N_1 \sim N_2 : \tau}{\Gamma \vdash M_1\, N_1 \sim M_2\, N_2 : \sigma}$$

C-If
$$\frac{\Gamma \vdash M_1 \sim M_2 : \mathsf{B} \qquad \Gamma \vdash N_1 \sim N_2 : \tau \qquad \Gamma \vdash N_1' \sim N_2' : \tau}{\Gamma \vdash \mathsf{if}\ M_1\ \mathsf{then}\ N_1\ \mathsf{else}\ N_1' \sim \mathsf{if}\ M_2\ \mathsf{then}\ N_2\ \mathsf{else}\ N_2' : \tau}$$

Fundamental lemma of logical equivalence

Theorem (Reflexivity) *(also called the fundamental lemma))*

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

Proof By induction on the typing derivation, using compatibility lemmas.

Compatibility lemmas

C-True
$$\Gamma \vdash \mathsf{tt} \sim \mathsf{tt} : \mathit{bool}$$

C-False
$$\Gamma \vdash \mathsf{ff} \sim \mathsf{ff} : \mathit{bool}$$

C-Var
$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x \sim x : \tau}$$

C-Abs
$$\frac{\Gamma, x : \tau \vdash M_1 \sim M_2 : \sigma}{\Gamma \vdash \lambda x{:}\tau.\, M_1 \sim \lambda x{:}\tau.\, M_2 : \tau \to \sigma}$$

C-App
$$\frac{\Gamma \vdash M_1 \sim M_2 : \tau \to \sigma \qquad \Gamma \vdash N_1 \sim N_2 : \tau}{\Gamma \vdash M_1\, N_1 \sim M_2\, N_2 : \sigma}$$

C-If
$$\frac{\Gamma \vdash M_1 \sim M_2 : \mathsf{B} \qquad \Gamma \vdash N_1 \sim N_2 : \tau \qquad \Gamma \vdash N_1' \sim N_2' : \tau}{\Gamma \vdash \mathsf{if}\ M_1\ \mathsf{then}\ N_1\ \mathsf{else}\ N_1' \sim \mathsf{if}\ M_2\ \mathsf{then}\ N_2\ \mathsf{else}\ N_2' : \tau}$$

Fundamental lemma of logical equivalence

Theorem (Reflexivity) *(also called the fundamental lemma))*

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

Proof By induction on the typing derivation, using compatibility lemmas.

Compatibility lemmas

$$\textsc{C-True}\quad \Gamma \vdash \mathsf{tt} \sim \mathsf{tt} : \mathit{bool}$$

$$\textsc{C-False}\quad \Gamma \vdash \mathsf{ff} \sim \mathsf{ff} : \mathit{bool}$$

$$\textsc{C-Var}\quad \frac{x : \tau \in \Gamma}{\Gamma \vdash x \sim x : \tau}$$

$$\textsc{C-Abs}\quad \frac{\Gamma, x : \tau \vdash M_1 \sim M_2 : \sigma}{\Gamma \vdash \lambda x{:}\tau.\, M_1 \sim \lambda x{:}\tau.\, M_2 : \tau \to \sigma}$$

$$\textsc{C-App}\quad \frac{\Gamma \vdash M_1 \sim M_2 : \tau \to \sigma \qquad \Gamma \vdash N_1 \sim N_2 : \tau}{\Gamma \vdash M_1\, N_1 \sim M_2\, N_2 : \sigma}$$

$$\textsc{C-If}\quad \frac{\Gamma \vdash M_1 \sim M_2 : \mathsf{B} \qquad \Gamma \vdash N_1 \sim N_2 : \tau \qquad \Gamma \vdash N_1' \sim N_2' : \tau}{\Gamma \vdash \mathsf{if}\ M_1\ \mathsf{then}\ N_1\ \mathsf{else}\ N_1' \sim \mathsf{if}\ M_2\ \mathsf{then}\ N_2\ \mathsf{else}\ N_2' : \tau}$$

Proof of compatibility lemmas

Each case can be shown independently.

Rule C-ABS: Assume $\Gamma, x : \tau \vdash M_1 \sim M_2 : \sigma$ $(\mathbf{1})$
We show $\Gamma \vdash \lambda x{:}\tau.\, M_1 \sim \lambda x{:}\tau.\, M_2 : \tau \to \sigma$. Let $\gamma \in \mathcal{G}[\![\Gamma]\!]$.
We show $(\gamma_1(\lambda x{:}\tau.\, M_1), \gamma_2(\lambda x{:}\tau.\, M_2)) \in \mathcal{V}[\![\tau \to \sigma]\!]$. Let (V_1, V_2) be in $\mathcal{V}[\![\tau]\!]$.
We show $(\gamma_1(\lambda x{:}\tau.\, M_1)\ V_1, \gamma_2(\lambda x{:}\tau.\, M_2)\ V_2) \in \mathcal{E}[\![\sigma]\!]$ $(\mathbf{2})$.

Since $\gamma_i(\lambda x{:}\tau.\, M_i)\ V_i \Downarrow (\gamma_i, x \mapsto V_i)M_i \triangleq \gamma'_i M_i$, by inverse reduction, it suffices to show $(\gamma'_1 M_1, \gamma'_2 M_2) \in \mathcal{E}[\![\sigma]\!]$. This follows from (1) since $\gamma' \in \mathcal{G}[\![\Gamma, x : \tau]\!]$.

Rule C-APP *(and* C-IF*)*: By induction hypothesis and the fact that substitution distributes over applications (and conditional).

We must show $\Gamma \vdash M_1\ N_1 \sim M_2\ M_2 : \sigma$ $(\mathbf{1})$. Let $\gamma \in \mathcal{G}[\![\Gamma]\!]$. From the premises $\Gamma \vdash M_1 \sim M_2 : \tau \to \sigma$ and $\Gamma \vdash N_1 \sim N_2 : \tau$, we have $(\gamma_1 M_1, \gamma_2 M_2) \in \mathcal{E}[\![\tau \to \sigma]\!]$ and $(\gamma_1 N_1, \gamma_2 N_2) \in \mathcal{E}[\![\tau]\!]$. Therefore $(\gamma_1 M_1\ \gamma_1 N_1, \gamma_2 M_2\ \gamma_2 N_2) \in \mathcal{E}[\![\sigma]\!]$. That is $(\gamma_1(M_1\ N_1), \gamma_2(M_2\ N_2)) \in \mathcal{E}[\![\sigma]\!]$, which proves (1).

Rule C-TRUE, C-FALSE, *and* C-VAR: are immediate

Proof of compatibility lemmas (cont.)

Rule C-IF: We show $\Gamma \vdash \text{if } M_1 \text{ then } N_1 \text{ else } N_1' \sim \text{if } M_2 \text{ then } N_2 \text{ else } N_2' : \tau$.
Assume $\gamma \in \mathcal{G}[\![\gamma]\!]$.
We show $(\gamma_1(\text{if } M_1 \text{ then } N_1 \text{ else } N_1'), \gamma_2(\text{if } M_2 \text{ then } N_2 \text{ else } N_2')) \in \mathcal{E}[\![\tau]\!]$, That is $(\text{if } \gamma_1 M_1 \text{ then } \gamma_1 N_1 \text{ else } \gamma_1 N_1', \text{if } \gamma_2 M_2 \text{ then } \gamma_2 N_2 \text{ else } \gamma_2 N_2') \in \mathcal{E}[\![\tau]\!]$ **(1)**.

From the premise $\Gamma \vdash M_1 \sim M_2 : \mathsf{B}$, we have $(\gamma_1 M_1, \gamma_2 M_2) \in \mathcal{E}[\![\mathsf{B}]\!]$. Therefore $M_1 \Downarrow V$ and $M_2 \Downarrow V$ where V is either tt or ff:

- *Case V is* tt:. Then, $(\text{if } \gamma_i M_i \text{ then } \gamma_i N_i \text{ else } \gamma_i N_i') \Downarrow \gamma_i N_i$, *i.e.* $\gamma_i(\text{if } M_i \text{ then } N_i \text{ else } N_i') \Downarrow \gamma_i N_i$. From the premise $\Gamma \vdash N_1 \sim N_2 : \tau$, we have $(\gamma_1 N_1, \gamma_2 N_2) \in \mathcal{E}[\![\tau]\!]$ and (1) follows by closer by inverse reduction.
- *Case V is* ff: similar.

Proof of reflexivity

By induction on the derivation of $\Gamma \vdash M : \tau$.
We must show $\Gamma \vdash M \sim M : \tau$:

All cases immediately follow from compatibility lemmas.

Case M *is* tt *or* ff: Immediate by Rule C-True or Rule C-False

Case M *is* x: Immediate by Rule C-Var.

Case M *is* $M'\ N$: By inversion of the typing rule App, induction hypothesis, and Rule C-App.

Case M *is* $\lambda\tau{:}N.$: By inversion of the typing rule Abs, induction hypothesis, and Rule C-Abs.

Properties of logical relations

Corollary (equivalence) Open logical relation is an equivalence relation

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—but *we need a syntactic definition of context-typing derivations* (which we have omitted) to be able to reason by induction on the context-typing derivation.

Soundness of logical equivalence
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 coarsiest such relation.

Properties of logical equivalence

Completeness of logical equivalence
Observational equivalence of closed terms implies logical equivalence.
That is $(\cong_\tau) \subseteq (\sim_\tau)$.

Proof by induction on τ.

Case B: In the empty context, by consistency $\cong_{\mathsf{B}}$ is a subrelation of $\simeq_{\mathsf{B}}$ which coincides with $\sim_{\mathsf{B}}$.

Case $\tau \to \sigma$: By congruence of observational equivalence!

By hypothesis, we have $M_1 \cong_{\tau\to\sigma} M_2$ $(\mathbf{1})$. To show $M_1 \sim_{\tau\to\sigma} M_2$, we assume $V_1 \approx_\tau V_2$ $(\mathbf{2})$ and show $M_1\, V_1 \sim_\sigma M_2\, V_2$ $(\mathbf{3})$.

By soundness applied to (2), we have $V_1 \cong_\tau V_2$ from (2). By congruence with (1), we have $M_1\, V_1 \cong_\sigma M_2\, V_2$, which implies (3) by IH at type σ.

Logical equivalence: example of application

Fact: Assume $not \triangleq \lambda x{:}\mathsf{B}.\, \text{if } x \text{ then ff else tt}$
and $M \triangleq \lambda x{:}\mathsf{B}.\, \lambda y{:}\tau.\, \lambda z{:}\tau.\, \text{if } not\ x \text{ then } y \text{ else } z$
and $M' \triangleq \lambda x{:}\mathsf{B}.\, \lambda y{:}\tau.\, \lambda z{:}\tau.\, \text{if } x \text{ then } z \text{ else } y$.

Show that $M \simeq_{\mathsf{B}\to\tau\to\tau\to\tau} M'$.

?

Logical equivalence: example of application

Fact: Assume $not \triangleq \lambda x{:}\mathsf{B}.\, \text{if } x \text{ then ff else tt}$
and $M \triangleq \lambda x{:}\mathsf{B}.\, \lambda y{:}\tau.\, \lambda z{:}\tau.\, \text{if } not\ x \text{ then } y \text{ else } z$
and $M' \triangleq \lambda x{:}\mathsf{B}.\, \lambda y{:}\tau.\, \lambda z{:}\tau.\, \text{if } x \text{ then } z \text{ else } y$.

Show that $M \cong_{\mathsf{B}\to\tau\to\tau\to\tau} M'$.

Proof

It suffices to show $M\ V_0\ V_1\ V_2 \sim_\tau M'\ V_0'\ V_1'\ V_2'$ whenever $V_0 \approx_{\mathsf{B}} V_0'$ $(\mathbf{1})$ and $V_1 \approx_\tau V_1'$ $(\mathbf{2})$ and $V_2 \approx_\tau V_2'$ $(\mathbf{3})$.

?

Logical equivalence: example of application

Fact: Assume $not \triangleq \lambda x{:}\mathsf{B}.\,$if x then ff else tt
and $M \triangleq \lambda x{:}\mathsf{B}.\,\lambda y{:}\tau.\,\lambda z{:}\tau.\,$if $not\ x$ then y else z
and $M' \triangleq \lambda x{:}\mathsf{B}.\,\lambda y{:}\tau.\,\lambda z{:}\tau.\,$if x then z else y.

Show that $M \cong_{\mathsf{B}\to\tau\to\tau\to\tau} M'$.

Proof

It suffices to show $M\ V_0\ V_1\ V_2 \sim_\tau M'\ V_0'\ V_1'\ V_2'$ whenever $V_0 \approx_{\mathsf{B}} V_0'$ $(\mathbf{1})$ and $V_1 \approx_\tau V_1'$ $(\mathbf{2})$ and $V_2 \approx_\tau V_2'$ $(\mathbf{3})$. By inverse reduction, it suffices to show: if $not\ V_0$ then V_1 else V_2 $\sim_\tau$ if V_0' then V_2' else V_1' $(\mathbf{4})$.

?

Logical equivalence: example of application

Fact: Assume $not \triangleq \lambda x{:}\mathsf{B}.\,\text{if } x \text{ then ff else tt}$
and $M \triangleq \lambda x{:}\mathsf{B}.\,\lambda y{:}\tau.\,\lambda z{:}\tau.\,\text{if } not\ x \text{ then } y \text{ else } z$
and $M' \triangleq \lambda x{:}\mathsf{B}.\,\lambda y{:}\tau.\,\lambda z{:}\tau.\,\text{if } x \text{ then } z \text{ else } y$.

Show that $M \cong_{\mathsf{B}\to\tau\to\tau\to\tau} M'$.

Proof

It suffices to show $M\ V_0\ V_1\ V_2 \sim_\tau M'\ V_0'\ V_1'\ V_2'$ whenever $V_0 \approx_{\mathsf{B}} V_0'$ $\mathbf{(1)}$ and $V_1 \approx_\tau V_1'$ $\mathbf{(2)}$ and $V_2 \approx_\tau V_2'$ $\mathbf{(3)}$. By inverse reduction, it suffices to show: if $not\ V_0$ then V_1 else V_2 $\sim_\tau$ if V_0' then V_2' else V_1' $\mathbf{(4)}$.

It follows from (1) that we have only two cases:

Case $V_0 = V_0' = \text{tt}$: Then $not\ V_0 \Downarrow \text{ff}$ and thus $M \Downarrow V_2$ while $M' \Downarrow V_2$. Then (4) follows by inverse reduction and (3).

Case $V_0 = V_0' = \text{ff}$: is symmetric.

Contents

Observational equivalence

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

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

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

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

Observational equivalence

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

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

As before, write $M \cong_\tau N$ for $\varnothing; \varnothing \vdash M \cong N : \tau$ (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 (sets of pairs of related values) of closed types ρ_1 and ρ_2

Let $\mathcal{R}(\rho_1, \rho_2)$ be the set of relations on values of closed types ρ_1 and ρ_2, that is $\mathcal{P}(\mathsf{Val}(\rho_1) \times \mathsf{Val}(\rho_2))$. We optionally restrict to *admissible* relations, *i.e.* relations that are *closed by observational equivalence*:

$$
\begin{aligned}
&R \in \mathcal{R}^{\sharp}(\tau_1, \tau_2) \implies \\
&\qquad \forall (V_1, V_2) \in R,\ \forall W_1, W_2,\ W_1 \cong V_1 \wedge W_2 \cong V_2 \implies (W_1, W_2) \in R
\end{aligned}
$$

The restriction to *admissible relations* is required for *completeness* of logical equivalence with respect to observational equivalence but *not for soundness*.

Example of admissible relations

For example, both

$$R_1 \triangleq \{(\mathsf{tt}, 0), (\mathsf{ff}, 1)\}$$
$$R_2 \triangleq \{(\mathsf{tt}, 0)\} \cup \{(\mathsf{ff}, n) \mid n \in \mathbb{Z}^\star\}$$

are admissible relations in $\mathcal{R}^\sharp(\mathsf{B}, \mathit{int})$.

But

$$R_3 \triangleq \{(\mathsf{tt}, \lambda x{:}\tau.\, 0), (\mathsf{ff}, \lambda x{:}\tau.\, 1)\}$$

although in $\mathcal{R}(\mathsf{B}, \tau \to \mathit{int})$, is not admissible.

Why?

Example of admissible relations

For example, both

$$R_1 \triangleq \{(\mathsf{tt}, 0), (\mathsf{ff}, 1)\}$$
$$R_2 \triangleq \{(\mathsf{tt}, 0)\} \cup \{(\mathsf{ff}, n) \mid n \in \mathbb{Z}^\star\}$$

are admissible relations in $\mathcal{R}^\sharp(\mathsf{B}, \mathit{int})$.

But

$$R_3 \triangleq \{(\mathsf{tt}, \lambda x{:}\tau.\, 0), (\mathsf{ff}, \lambda x{:}\tau.\, 1)\}$$

although in $\mathcal{R}(\mathsf{B}, \tau \to \mathit{int})$, is not admissible.

Taking $M_0 \triangleq \lambda x{:}\tau.\, (\lambda z{:}\mathit{int}.\, z)\, 0$, we have $M \cong_{\tau\to\mathit{int}} \lambda x{:}\tau.\, 0$ but (tt, M) is not in R_3.

Example of admissible relations

For example, both

$$R_1 \triangleq \{(\mathsf{tt}, 0), (\mathsf{ff}, 1)\}$$
$$R_2 \triangleq \{(\mathsf{tt}, 0)\} \cup \{(\mathsf{ff}, n) \mid n \in \mathbb{Z}^\star\}$$

are admissible relations in $\mathcal{R}^\sharp(\mathsf{B}, \mathit{int})$.

But

$$R_3 \triangleq \{(\mathsf{tt}, \lambda x\!:\!\tau.\, 0), (\mathsf{ff}, \lambda x\!:\!\tau.\, 1)\}$$

although in $\mathcal{R}(\mathsf{B}, \tau \to \mathit{int})$, is not admissible.

Note A relation R in $\mathcal{R}(\tau_1, \tau_2)$ can always be turned into an admissible relation $R^\sharp$ in $\mathcal{R}^\sharp(\tau_1, \tau_2)$ by closing R by observational equivalence.

Note It is *a key* that such relations can relate values at different types.

Interpretation of type environments

Interpretation of type variables

We write η for mappings $\alpha \mapsto (\rho_1, \rho_2, R)$ where $R \in \mathcal{R}(\rho_1, \rho_2)$.

We write η_i (*resp.* η_R) for the type (*resp.* relational) substitution that maps α to ρ_i (*resp.* R) whenever η maps α to (ρ_1, ρ_2, R).

We define

$$\mathcal{V}[\![\alpha]\!]_\eta \triangleq \eta_R(\alpha)$$

$$\mathcal{V}[\![\forall\alpha.\,\tau]\!]_\eta \triangleq \{(V_1, V_2) \mid V_1 : \eta_1(\forall\alpha.\,\tau) \wedge V_2 : \eta_2(\forall\alpha.\,\tau) \wedge \\ \forall \rho_1, \rho_2, \forall R \in \mathcal{R}(\rho_1, \rho_2), (V_1\ \rho_1, V_2\ \rho_2) \in \mathcal{E}[\![\tau]\!]_{\eta, \alpha \mapsto (\rho_1, \rho_2, R)}\}$$

Logical equivalence for closed terms with open types

We redefine

$$
\begin{aligned}
\mathcal{V}[\![\mathsf{B}]\!]_\eta &\triangleq \{(\mathsf{tt},\mathsf{tt}),(\mathsf{ff},\mathsf{ff})\}\\
\mathcal{V}[\![\tau \to \sigma]\!]_\eta &\triangleq \{(V_1,V_2) \mid V_1 \vdash \eta_1(\tau\to\sigma) \wedge V_2 \vdash \eta_2(\tau\to\sigma) \wedge\\
&\qquad \forall (W_1,W_2) \in \mathcal{V}[\![\tau]\!]_\eta,\ (V_1\ W_1, V_2\ W_2) \in \mathcal{E}[\![\sigma]\!]_\eta\}\\
\mathcal{E}[\![\tau]\!]_\eta &\triangleq \{(M_1,M_2) \mid M_1 : \eta_1\tau \wedge M_2 : \eta_2\tau \wedge\\
&\qquad \exists (V_1,V_2) \in \mathcal{V}[\![\tau]\!]_\eta, M_1 \Downarrow V_1 \wedge M_2 \Downarrow V_2\}\\
\mathcal{G}[\![\varnothing]\!]_\eta &\triangleq \{\varnothing\}\\
\mathcal{G}[\![\Gamma, x:\tau]\!]_\eta &\triangleq \{\gamma, x \mapsto (V_1,V_2) \mid \gamma \in \mathcal{G}[\![\Gamma]\!]_\eta \wedge (V_1,V_2) \in \mathcal{V}[\![\tau]\!]_\eta\}
\end{aligned}
$$

and define

$$
\begin{aligned}
\mathcal{D}[\![\varnothing]\!] &\triangleq \{\varnothing\}\\
\mathcal{D}[\![\Delta,\alpha]\!] &\triangleq \{\eta, \alpha \mapsto (\rho_1,\rho_2,\mathcal{R}) \mid \eta \in \mathcal{D}[\![\Delta]\!] \wedge R \in \mathcal{R}(\rho_1,\rho_2)\}
\end{aligned}
$$

Logical equivalence for open terms

Definition We define $\Delta; \Gamma \vdash M \sim M' : \tau$ as

$$\wedge \left\{ \begin{array}{l} \Delta; \Gamma \vdash M, M' : \tau \\ \forall \eta \in \mathcal{D}[\![\Delta]\!],\ \forall \gamma \in \mathcal{G}[\![\Gamma]\!]_\eta,\ (\eta_1(\gamma_1 M_1), \eta_2(\gamma_2 M_2)) \in \mathcal{E}[\![\tau]\!]_\eta \end{array} \right.$$

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

Notation

We also write $M_1 \sim_\tau M_2$ for $\vdash M_1 \sim M_2 : \tau$ (*i.e.* $\varnothing; \varnothing \vdash M_1 \sim M_2 : \tau$).

In this case, τ is a closed type and M_1 and M_2 are closed terms of type τ; hence, this coincides with the previous definition (M_1, M_2) in $\mathcal{E}[\![\tau]\!]$, which may still be used as a shorthand for $\mathcal{E}[\![\tau]\!]_\varnothing$.

Properties

Respect for observational equivalence

If $(M_1, M_2) \in \mathcal{E}[\![\tau]\!]^\sharp_\eta$ and $N_1 \cong_{\eta_1(\tau)} M_1$ and $N_2 \cong_{\eta_2(\tau)} M_2$ then $(N_1, N_2) \in \mathcal{E}[\![\tau]\!]^\sharp_\eta$.

Requires admissibility

(We use $^\sharp$ to indicate that admissibility is required in the definition of $\mathcal{R}^\sharp$)

Proof. By induction on τ.

Assume $(M_1, M_2) \in \mathcal{E}[\![\tau]\!]_\eta$ $(\mathbf{1})$ and $N_1 \cong_{\eta_1(\tau)} M_1$ $(\mathbf{2})$. We show $(N_1, M_2) \in \mathcal{E}[\![\tau]\!]_\eta$.

Case τ is $\forall\alpha.\sigma$: Assume $R \in \mathcal{R}^\sharp(\rho_1, \rho_2)$. Let η_α be $\eta, \alpha \mapsto (\rho_1, \rho_2, R)$.
We have $(M_1\ \rho_1, M_2\ \rho_2) \in \mathcal{E}[\![\sigma]\!]_{\eta_\alpha}$, from (1).
By congruence from (2), we have $N_1 \rho_1 \cong_{\delta(\tau)} M_1\ \rho_1$.
Hence, by induction hypothesis, $(M_1\ \rho_1, M_2\ \rho_2) \in \mathcal{E}[\![\sigma]\!]_{\eta_\alpha}$, as expected.

Case τ is α: Relies on admissibility, indeed.

Other cases: the proof is similar to the case of the simply-typed λ-calculus.

Properties

Respect for observational equivalence

If $(M_1, M_2) \in \mathcal{E}[\![\tau]\!]^\sharp_\eta$ and $N_1 \cong_{\eta_1(\tau)} M_1$ and $N_2 \cong_{\eta_2(\tau)} M_2$ then $(N_1, N_2) \in \mathcal{E}[\![\tau]\!]^\sharp_\eta$. Requires admissibility

(We use $^\sharp$ to indicate that admissibility is required in the definition of $\mathcal{R}^\sharp$)

Proof. By induction on τ.

Corollary

The relation $\mathcal{V}[\![\tau]\!]^\sharp_\eta$ is an admissible relation in $\mathcal{R}^\sharp(\eta_1\tau, \eta_2\tau)$.

Application: we may take this relation when admissibility is required.

Properties

Lemma (Closure under observational equivalence)
If $\Delta; \Gamma \vdash M_1 \sim^\sharp M_2 : \tau$ and $\Delta; \Gamma \vdash M_1 \cong N_1 : \tau$ and $\Delta; \Gamma \vdash M_2 \cong N_2 : \tau$, then $\Delta; \Gamma \vdash N_1 \sim^\sharp N_2 : \tau$ Requires admissibility

Lemma (Compositionality) Key lemma

Assume $\Delta \vdash \sigma$ and $\Delta, \alpha \vdash \tau$ and $\eta \in \mathcal{D}[\![\Delta]\!]$. Then,

$$\mathcal{V}[\![\tau[\alpha \mapsto \sigma]]\!]_\eta \quad = \quad \mathcal{V}[\![\tau]\!]_{\eta, \alpha \mapsto (\eta_1 \sigma, \eta_2 \sigma, \mathcal{V}[\![\sigma]\!]_\eta)}$$

Proof by induction on τ.

Parametricity

Theorem (Reflexivity) *(also called the fundamental lemma)*

If $\Delta; \Gamma \vdash M : \tau$ then $\Delta; \Gamma \vdash M \sim M : \tau$.

Notice: Admissibility is not required for the fundamental lemma

Proof by induction on the typing derivation, using compatibility lemmas.

Compatibility lemmas

We redefine the lemmas to work in a typing context of the form Δ, Γ instead of Γ and add two new lemmas:

C-Tabs
$$\frac{\Delta, \alpha; \Gamma \vdash M_1 \sim M_2 : \tau}{\Delta; \Gamma \vdash \Lambda\alpha.\, M_1 \sim \Lambda\alpha.\, M_2 : \forall\alpha.\, \tau}$$

C-Tapp
$$\frac{\Delta; \Gamma \vdash M_1 \sim M_2 : \forall\alpha.\, \tau \qquad \Delta \vdash \sigma}{\Delta; \Gamma \vdash M_1\ \sigma \sim M_2\ \sigma : \tau[\alpha \mapsto \sigma]}$$

Properties

Soundness of logical equivalence
Logical equivalence implies observational equivalence.
If $\Delta; \Gamma \vdash M_1 \sim M_2 : \tau$ then $\Delta; \Gamma \vdash M_1 \cong M_2 : \tau$.

Completeness of logical equivalence
Observational equivalence implies logical equivalence with admissibility.
If $\Delta; \Gamma \vdash M_1 \cong M_2 : \tau$ then $\Delta; \Gamma \vdash M_1 \sim^{\sharp} M_2 : \tau$.

As a particular case, $M_1 \cong_\tau M_2$ iff $M_1 \sim^{\sharp}_\tau M_2$.

Note: Admissibility is not required for soundness—only for completeness.

That is, proofs that some observational equivalence hold do not usually require admissibility.

Properties

Extensionality *(A fact, hence does not depend on admissibility)*

$M_1 \cong_{\tau\to\sigma} M_2$ iff $\forall (V:\tau), M_1\ V \cong_\sigma M_2\ V$ iff $\forall (N:\tau), M_1\ N \cong_\sigma M_2\ N$

$M_1 \cong_{\forall\alpha.\,\tau} M_2$ iff for all closed type ρ, $M_1\ \rho \cong_{\tau[\alpha\mapsto\rho]} M_2\ \rho$.

Proof. Forward direction is immediate as $\cong$ is a congruence. Backward direction uses logical relations and admissibility, but the exported statement does not.

Case Value abstraction: It suffices to show $M_1 \sim_{\tau\to\sigma} M_2$. That is, assuming $N_1 \sim_\tau N_2$ $(\mathbf{1})$, we show $M_1\ N_1 \sim_\sigma M_2\ N_2$ $(\mathbf{2})$. By assumption, we have $M_1\ N_1 \cong_\sigma M_2\ N_1$ $(\mathbf{3})$. By the fundamental lemma, we have $M_2 \sim_{\tau\to\sigma} M_2$. Hence, from (1), we must have $M_2\ N_1 \sim_\sigma M_2\ N_2$, We conclude (2) by *respect for observational equivalence* with (3)—which requires admissibility.

Case Type abstraction: It suffices to show $M_1 \sim_{\forall\alpha.\,\tau} M_2$. That is, given $R \in \mathcal{R}(\rho_1, \rho_2)$, we show $(M_1\ \rho_1, M_2\ \rho_2) \in \mathcal{E}[\![\tau]\!]_{\alpha\mapsto(\rho_1,\rho_2,R)}$ $(\mathbf{4})$.
By assumption, we have $M_1\ \rho_1 \cong_{\tau[\alpha\mapsto\rho_1]} M_2\ \rho_1$ $(\mathbf{5})$.
By the fundamental lemma, we have $M_2 \sim_{\forall\alpha.\,\tau} M_2$.
Hence, we have $(M_2\ \rho_1, M_2\ \rho_2) \in \mathcal{E}[\![\tau]\!]_{\alpha\mapsto(\rho_1,\rho_2,R)}$
We conclude (4) by *respect for observational equivalence* with (5).

Contents

Applications Inhabitants of $\forall\alpha.\,\alpha \to \alpha$

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

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

By parametricity, we have $M \sim_{\forall\alpha.\,\alpha\to\alpha} M$ $(\mathbf{2})$.

Consider R in $\mathcal{R}(\rho,\rho)$ equal to $\{(V,V)\}$ and η be $[\alpha \mapsto (\rho,\rho,R)]$. (3)
By construction, we have $(V,V) \in \mathcal{V}[\![\alpha]\!]_\eta$.

Hence, from (2), we have $(M\ \rho\ V, M\ \rho\ V) \in \mathcal{E}[\![\alpha]\!]_\eta$, which means that the pair $(M\ \rho\ V, M\ \rho\ V)$ reduces to a pair of values in (the singleton) R. This implies that $M\ \rho\ V$ reduces to V, which in turn, implies (1).

$(\mathbf{3})$ Admissibility is not needed

Applications

Inhabitants of $\forall\alpha.\,\alpha \to \alpha \to \alpha$

Fact Let σ be $\forall\alpha.\,\alpha \to \alpha \to \alpha$. If $M : \sigma$, then either
$M \cong_\sigma W_1 \triangleq \Lambda\alpha.\,\lambda x_1{:}\alpha.\,\lambda x_2{:}\alpha.\,x_1$ or $M \cong_\sigma W_2 \triangleq \Lambda\alpha.\,\lambda x_1{:}\alpha.\,\lambda x_2{:}\alpha.\,x_2$

Proof By *extensionality*, it suffices to show that for either $i = 1$ or $i = 2$, for any closed type ρ and $V_1, V_2 : \rho$, we have $M\ \rho\ V_1\ V_2 \cong_\rho W_i\ \rho\ V_1\ V_2$, or just $M\ \rho\ V_1\ V_2 \cong_\sigma V_i$ $(\mathbf{1})$.

?

Applications Inhabitants of $\forall\alpha.\,\alpha \to \alpha \to \alpha$

Fact Let σ be $\forall\alpha.\,\alpha \to \alpha \to \alpha$. If $M : \sigma$, then either
$M \cong_\sigma W_1 \triangleq \Lambda\alpha.\,\lambda x_1{:}\alpha.\,\lambda x_2{:}\alpha.\,x_1$ or $M \cong_\sigma W_2 \triangleq \Lambda\alpha.\,\lambda x_1{:}\alpha.\,\lambda x_2{:}\alpha.\,x_2$

Proof By *extensionality*, it suffices to show that for either $i = 1$ or $i = 2$, for any closed type ρ and $V_1, V_2 : \rho$, we have $M\ \rho\ V_1\ V_2 \cong_\rho W_i\ \rho\ V_1\ V_2$, or just $M\ \rho\ V_1\ V_2 \cong_\sigma V_i$ $(\mathbf{1})$.

Let ρ and $V_1, V_2 : \rho$ be fixed. Consider R equal to $\{(\mathsf{tt}, V_1), (\mathsf{ff}, V_2)\}$ in $\mathcal{R}(\mathsf{B}, \rho)$ and η be $\alpha \mapsto (\mathsf{B}, \rho, R)$.

?

Applications — Inhabitants of $\forall\alpha.\,\alpha \to \alpha \to \alpha$

Fact Let σ be $\forall\alpha.\,\alpha \to \alpha \to \alpha$. If $M : \sigma$, then either
$M \cong_\sigma W_1 \triangleq \Lambda\alpha.\,\lambda x_1{:}\alpha.\,\lambda x_2{:}\alpha.\,x_1$ or $M \cong_\sigma W_2 \triangleq \Lambda\alpha.\,\lambda x_1{:}\alpha.\,\lambda x_2{:}\alpha.\,x_2$

Proof By *extensionality*, it suffices to show that for either $i = 1$ or $i = 2$, for any closed type ρ and $V_1, V_2 : \rho$, we have $M\ \rho\, V_1\ V_2 \cong_\rho W_i\ \rho\, V_1\ V_2$, or just $M\ \rho\, V_1\ V_2 \cong_\sigma V_i$ $(\mathbf{1})$.

Let ρ and $V_1, V_2 : \rho$ be fixed. Consider R equal to $\{(\mathsf{tt}, V_1), (\mathsf{ff}, V_2)\}$ in $\mathcal{R}(\mathsf{B}, \rho)$ and η be $\alpha \mapsto (\mathsf{B}, \rho, R)$. We have $(\mathsf{tt}, V_1) \in \mathcal{V}[\![\alpha]\!]_\eta$ since $R(\mathsf{tt}, V_1)$ and, similarly, $(\mathsf{ff}, V_2) \in \mathcal{V}[\![\alpha]\!]_\eta$.

We have $(M, M) \in \mathcal{E}[\![\sigma]\!]$ by parametricity.

?

Applications — Inhabitants of $\forall\alpha.\,\alpha \to \alpha \to \alpha$

Fact Let σ be $\forall\alpha.\,\alpha \to \alpha \to \alpha$. If $M : \sigma$, then either
$M \cong_\sigma W_1 \triangleq \Lambda\alpha.\,\lambda x_1{:}\alpha.\,\lambda x_2{:}\alpha.\,x_1$ or $M \cong_\sigma W_2 \triangleq \Lambda\alpha.\,\lambda x_1{:}\alpha.\,\lambda x_2{:}\alpha.\,x_2$

Proof By *extensionality*, it suffices to show that for either $i = 1$ or $i = 2$, for any closed type ρ and $V_1, V_2 : \rho$, we have $M\ \rho\ V_1\ V_2 \cong_\rho W_i\ \rho\ V_1\ V_2$, or just $M\ \rho\ V_1\ V_2 \cong_\sigma V_i$ $(\mathbf{1})$.

Let ρ and $V_1, V_2 : \rho$ be fixed. Consider R equal to $\{(\mathsf{tt}, V_1), (\mathsf{ff}, V_2)\}$ in $\mathcal{R}(\mathsf{B}, \rho)$ and η be $\alpha \mapsto (\mathsf{B}, \rho, R)$. We have $(\mathsf{tt}, V_1) \in \mathcal{V}[\![\alpha]\!]_\eta$ since $R(\mathsf{tt}, V_1)$ and, similarly, $(\mathsf{ff}, V_2) \in \mathcal{V}[\![\alpha]\!]_\eta$.

We have $(M, M) \in \mathcal{E}[\![\sigma]\!]$ by parametricity. Hence, $(M\ \mathsf{B}\ \mathsf{tt}\ \mathsf{ff}, M\ \rho\ V_1\ V_2)$ is in $\mathcal{V}[\![\alpha]\!]_\eta$, which means that $(M\ \mathsf{B}\ \mathsf{tt}\ \mathsf{ff}, M\ \rho\ V_1\ V_2)$ reduces to a pair of values in R, which implies:

$$\bigvee \begin{cases} M\ \mathsf{B}\ \mathsf{tt}\ \mathsf{ff} \cong_{\mathsf{B}} \mathsf{tt} & \wedge \quad M\ \rho\ V_1\ V_2 \cong_\rho V_1 \\ M\ \mathsf{B}\ \mathsf{tt}\ \mathsf{ff} \cong_{\mathsf{B}} \mathsf{ff} & \wedge \quad M\ \rho\ V_1\ V_2 \cong_\rho V_2 \end{cases}$$

Next ?

Applications — Inhabitants of $\forall\alpha.\,\alpha \to \alpha \to \alpha$

Fact Let σ be $\forall\alpha.\,\alpha \to \alpha \to \alpha$. If $M : \sigma$, then either
$M \cong_\sigma W_1 \triangleq \Lambda\alpha.\,\lambda x_1{:}\alpha.\,\lambda x_2{:}\alpha.\,x_1$ or $M \cong_\sigma W_2 \triangleq \Lambda\alpha.\,\lambda x_1{:}\alpha.\,\lambda x_2{:}\alpha.\,x_2$

Proof By *extensionality*, it suffices to show that for either $i = 1$ or $i = 2$, for any closed type ρ and $V_1, V_2 : \rho$, we have $M\ \rho\ V_1\ V_2 \cong_\rho W_i\ \rho\ V_1\ V_2$, or just $M\ \rho\ V_1\ V_2 \cong_\sigma V_i$ (**1**).

Let ρ and $V_1, V_2 : \rho$ be fixed. Consider R equal to $\{(\mathsf{tt}, V_1), (\mathsf{ff}, V_2)\}$ in $\mathcal{R}(\mathsf{B}, \rho)$ and η be $\alpha \mapsto (\mathsf{B}, \rho, R)$. We have $(\mathsf{tt}, V_1) \in \mathcal{V}[\![\alpha]\!]_\eta$ since $R(\mathsf{tt}, V_1)$ and, similarly, $(\mathsf{ff}, V_2) \in \mathcal{V}[\![\alpha]\!]_\eta$.

We have $(M, M) \in \mathcal{E}[\![\sigma]\!]$ by parametricity. Hence, $(M\ \mathsf{B}\ \mathsf{tt}\ \mathsf{ff}, M\ \rho\ V_1\ V_2)$ is in $\mathcal{V}[\![\alpha]\!]_\eta$, which means that $(M\ \mathsf{B}\ \mathsf{tt}\ \mathsf{ff}, M\ \rho\ V_1\ V_2)$ reduces to a pair of values in R, which implies:

$$\forall \rho, V_1, V_2, \quad \bigvee \begin{cases} M\ \mathsf{B}\ \mathsf{tt}\ \mathsf{ff} \cong_{\mathsf{B}} \mathsf{tt} & \wedge \quad M\ \rho\ V_1\ V_2 \cong_\rho V_1 \\ M\ \mathsf{B}\ \mathsf{tt}\ \mathsf{ff} \cong_{\mathsf{B}} \mathsf{ff} & \wedge \quad M\ \rho\ V_1\ V_2 \cong_\rho V_2 \end{cases}$$

Since, M B tt ff is independent of ρ, V_1, and V_2, this actually shows (1).

Applications — Inhabitants of $\forall\alpha.\,\alpha \to \alpha \to \alpha$

Fact Let σ be $\forall\alpha.\,\alpha \to \alpha \to \alpha$. If $M : \sigma$, then either
$M \cong_\sigma W_1 \triangleq \Lambda\alpha.\,\lambda x_1{:}\alpha.\,\lambda x_2{:}\alpha.\,x_1$ or $M \cong_\sigma W_2 \triangleq \Lambda\alpha.\,\lambda x_1{:}\alpha.\,\lambda x_2{:}\alpha.\,x_2$

Proof By *extensionality*, it suffices to show that for either $i = 1$ or $i = 2$, for any closed type ρ and $V_1, V_2 : \rho$, we have $M\ \rho\ V_1\ V_2 \cong_\rho W_i\ \rho\ V_1\ V_2$, or just $M\ \rho\ V_1\ V_2 \cong_\sigma V_i$ $(\mathbf{1})$.

Let ρ and $V_1, V_2 : \rho$ be fixed. Consider R equal to $\{(\mathsf{tt}, V_1), (\mathsf{ff}, V_2)\}$ in $\mathcal{R}(\mathsf{B}, \rho)$ and η be $\alpha \mapsto (\mathsf{B}, \rho, R)$. We have $(\mathsf{tt}, V_1) \in \mathcal{V}[\![\alpha]\!]_\eta$ since $R(\mathsf{tt}, V_1)$ and, similarly, $(\mathsf{ff}, V_2) \in \mathcal{V}[\![\alpha]\!]_\eta$.

We have $(M, M) \in \mathcal{E}[\![\sigma]\!]$ by parametricity. Hence, $(M\ \mathsf{B}\ \mathsf{tt}\ \mathsf{ff}, M\ \rho\ V_1\ V_2)$ is in $\mathcal{V}[\![\alpha]\!]_\eta$, which means that $(M\ \mathsf{B}\ \mathsf{tt}\ \mathsf{ff}, M\ \rho\ V_1\ V_2)$ reduces to a pair of values in R, which implies:

$$\bigvee \begin{cases} \forall \rho, V_1, V_2,\ \ M\ \mathsf{B}\ \mathsf{tt}\ \mathsf{ff} \cong_{\mathsf{B}} \mathsf{tt} \ \ \wedge\ \ M\ \rho\ V_1\ V_2 \cong_\rho V_1 \\ \forall \rho, V_1, V_2,\ \ M\ \mathsf{B}\ \mathsf{tt}\ \mathsf{ff} \cong_{\mathsf{B}} \mathsf{ff} \ \ \wedge\ \ M\ \rho\ V_1\ V_2 \cong_\rho V_2 \end{cases}$$

Since, $M\ \mathsf{B}\ \mathsf{tt}\ \mathsf{ff}$ is independent of ρ, V_1, and V_2, this actually shows (1).

Applications — Inhabitants of $\forall\alpha.\,\alpha \to \alpha \to \alpha$

Fact Let σ be $\forall\alpha.\,\alpha \to \alpha \to \alpha$. If $M : \sigma$, then either
$M \cong_\sigma W_1 \triangleq \Lambda\alpha.\,\lambda x_1{:}\alpha.\,\lambda x_2{:}\alpha.\,x_1$ or $M \cong_\sigma W_2 \triangleq \Lambda\alpha.\,\lambda x_1{:}\alpha.\,\lambda x_2{:}\alpha.\,x_2$

Proof By *extensionality*, it suffices to show that for either $i = 1$ or $i = 2$, for any closed type ρ and $V_1, V_2 : \rho$, we have $M\ \rho\ V_1\ V_2 \cong_\rho W_i\ \rho\ V_1\ V_2$, or just $M\ \rho\ V_1\ V_2 \cong_\sigma V_i$ $(\mathbf{1})$.

Let ρ and $V_1, V_2 : \rho$ be fixed. Consider R equal to $\{(\text{tt}, V_1), (\text{ff}, V_2)\}$ in $\mathcal{R}(\text{B}, \rho)$ and η be $\alpha \mapsto (\text{B}, \rho, R)$. We have $(\text{tt}, V_1) \in \mathcal{V}[\![\alpha]\!]_\eta$ since $R(\text{tt}, V_1)$ and, similarly, $(\text{ff}, V_2) \in \mathcal{V}[\![\alpha]\!]_\eta$.

We have $(M, M) \in \mathcal{E}[\![\sigma]\!]$ by parametricity. Hence, $(M\ \text{B tt ff}, M\ \rho\ V_1\ V_2)$ is in $\mathcal{V}[\![\alpha]\!]_\eta$, which means that $(M\ \text{B tt ff}, M\ \rho\ V_1\ V_2)$ reduces to a pair of values in R, which implies:

$$\bigvee \begin{cases} M\ \text{B tt ff} \cong_{\text{B}} \text{tt} & \wedge \quad M\ \rho\ V_1\ V_2 \cong_\rho V_1 \\ M\ \text{B tt ff} \cong_{\text{B}} \text{ff} & \wedge \quad M\ \rho\ V_1\ V_2 \cong_\rho V_2 \end{cases}$$

Since, M B tt ff is independent of ρ, V_1, and V_2, this actually shows (1).

Applications — Inhabitants of $\forall\alpha.\,\alpha \to \alpha \to \alpha$

Fact Let σ be $\forall\alpha.\,\alpha \to \alpha \to \alpha$. If $M : \sigma$, then either
$M \cong_\sigma W_1 \triangleq \Lambda\alpha.\,\lambda x_1{:}\alpha.\,\lambda x_2{:}\alpha.\,x_1$ or $M \cong_\sigma W_2 \triangleq \Lambda\alpha.\,\lambda x_1{:}\alpha.\,\lambda x_2{:}\alpha.\,x_2$

Proof By *extensionality*, it suffices to show that for either $i = 1$ or $i = 2$, for any closed type ρ and $V_1, V_2 : \rho$, we have $M\ \rho\ V_1\ V_2 \cong_\rho W_i\ \rho\ V_1\ V_2$, or just $M\ \rho\ V_1\ V_2 \cong_\sigma V_i$ $(\mathbf{1})$.

Let ρ and $V_1, V_2 : \rho$ be fixed. Consider R equal to $\{(\text{tt}, V_1), (\text{ff}, V_2)\}$ in $\mathcal{R}(\text{B}, \rho)$ and η be $\alpha \mapsto (\text{B}, \rho, R)$. We have $(\text{tt}, V_1) \in \mathcal{V}[\![\alpha]\!]_\eta$ since $R(\text{tt}, V_1)$ and, similarly, $(\text{ff}, V_2) \in \mathcal{V}[\![\alpha]\!]_\eta$.

We have $(M, M) \in \mathcal{E}[\![\sigma]\!]$ by parametricity. Hence, $(M\ \text{B}\ \text{tt}\ \text{ff}, M\ \rho\ V_1\ V_2)$ is in $\mathcal{V}[\![\alpha]\!]_\eta$, which means that $(M\ \text{B}\ \text{tt}\ \text{ff}, M\ \rho\ V_1\ V_2)$ reduces to a pair of values in R, which implies:

$$\bigvee \begin{cases} M\ \text{B}\ \text{tt}\ \text{ff} \cong_{\text{B}} \text{tt} & \wedge\quad M\ \rho\ V_1\ V_2 \cong_\rho V_1 \\ M\ \text{B}\ \text{tt}\ \text{ff} \cong_{\text{B}} \text{ff} & \wedge\quad M\ \rho\ V_1\ V_2 \cong_\rho V_2 \end{cases}$$

Since, $M\ \text{B}\ \text{tt}\ \text{ff}$ is independent of ρ, V_1, and V_2, this actually shows (1).

Applications

Inhabitants of $\forall\alpha.\,\alpha \to \alpha \to \alpha$

Fact Let σ be $\forall\alpha.\,\alpha \to \alpha \to \alpha$. If $M : \sigma$, then either
$M \cong_\sigma W_1 \triangleq \Lambda\alpha.\,\lambda x_1{:}\alpha.\,\lambda x_2{:}\alpha.\,x_1$ or $M \cong_\sigma W_2 \triangleq \Lambda\alpha.\,\lambda x_1{:}\alpha.\,\lambda x_2{:}\alpha.\,x_2$

Proof By *extensionality*, it suffices to show that for either $i = 1$ or $i = 2$, for any closed type ρ and $V_1, V_2 : \rho$, we have $M\ \rho\ V_1\ V_2 \cong_\rho W_i\ \rho\ V_1\ V_2$, or just $M\ \rho\ V_1\ V_2 \cong_\sigma V_i$ $(\mathbf{1})$.

Let ρ and $V_1, V_2 : \rho$ be fixed. Consider R equal to $\{(\mathbf{0}, V_1), (\mathbf{1}, V_2)\}$ in $\mathcal{R}(\mathbb{N}, \rho)$ and η be $\alpha \mapsto (\mathbb{N}, \rho, R)$. We have $(\mathbf{0}, V_1) \in \mathcal{V}[\![\alpha]\!]_\eta$ since $R(\mathbf{0}, V_1)$ and, similarly, $(\mathbf{1}, V_2) \in \mathcal{V}[\![\alpha]\!]_\eta$.

We have $(M, M) \in \mathcal{E}[\![\sigma]\!]$ by parametricity. Hence, $(M\ \mathbb{N}\ \mathbf{0}\ \mathbf{1}, M\ \rho\ V_1\ V_2)$ is in $\mathcal{V}[\![\alpha]\!]_\eta$, which means that $(M\ \mathbb{N}\ \mathbf{0}\ \mathbf{1}, M\ \rho\ V_1\ V_2)$ reduces to a pair of values in R, which implies:

$$\bigvee \begin{cases} M\ \mathbb{N}\ \mathbf{0}\ \mathbf{1} \cong_{\mathbb{N}} \mathbf{0} & \wedge\ M\ \rho\ V_1\ V_2 \cong_\rho V_1 \\ M\ \mathbb{N}\ \mathbf{0}\ \mathbf{1} \cong_{\mathbb{N}} \mathbf{1} & \wedge\ M\ \rho\ V_1\ V_2 \cong_\rho V_2 \end{cases}$$

Since, $M\ \mathbb{N}\ \mathbf{0}\ \mathbf{1}$ is independent of ρ, V_1, and V_2, this actually shows (1).

Applications

Inhabitants of $\forall\alpha.\, \alpha \to \alpha \to \alpha$

Fact Let σ be $\forall\alpha.\, \alpha \to \alpha \to \alpha$. If $M : \sigma$, then either
$M \cong_\sigma W_1 \triangleq \Lambda\alpha.\, \lambda x_1{:}\alpha.\, \lambda x_2{:}\alpha.\, x_1$ or $M \cong_\sigma W_2 \triangleq \Lambda\alpha.\, \lambda x_1{:}\alpha.\, \lambda x_2{:}\alpha.\, x_2$

Proof By *extensionality*, it suffices to show that for either $i = 1$ or $i = 2$, for any closed type ρ and $V_1, V_2 : \rho$, we have $M\ \rho\ V_1\ V_2 \cong_\rho W_i\ \rho\ V_1\ V_2$, or just $M\ \rho\ V_1\ V_2 \cong_\sigma V_i$ (**1**).

Let ρ and $V_1, V_2 : \rho$ be fixed. Consider R equal to $\{(W_1, V_1), (W_2, V_2)\}$ in $\mathcal{R}(\sigma\ \ , \rho)$ and η be $\alpha \mapsto (\sigma\ \ , \rho, R)$. We have $(W_1, V_1) \in \mathcal{V}[\![\alpha]\!]_\eta$ since $R(W_1, V_1)$ and, similarly, $(W_2, V_2) \in \mathcal{V}[\![\alpha]\!]_\eta$.

We have $(M, M) \in \mathcal{E}[\![\sigma]\!]$ by parametricity. Hence, $(M\ \sigma\ \ \ W_1\ W_2, M\ \rho\ V_1\ V_2)$ is in $\mathcal{V}[\![\alpha]\!]_\eta$, which means that $(M\ \sigma\ \ \ W_1\ W_2, M\ \rho\ V_1\ V_2)$ reduces to a pair of values in R, which implies:

$$\bigvee \begin{cases} M\ \sigma\ \ \ W_1\ W_2 \cong_\sigma\ \ W_1 \ \wedge\ M\ \rho\ V_1\ V_2 \cong_\rho V_1 \\ M\ \sigma\ \ \ W_1\ W_2 \cong_\sigma\ \ W_2 \ \wedge\ M\ \rho\ V_1\ V_2 \cong_\rho V_2 \end{cases}$$

Since, $M\ \sigma\ \ \ W_1\ W_2$ is independent of ρ, V_1, and V_2, this actually shows (1).

Applications — Inhabitants of $\forall\alpha.\,\alpha \to \alpha \to \alpha$

Fact Let σ be $\forall\alpha.\,\alpha \to \alpha \to \alpha$. If $M : \sigma$, then either
$M \cong_\sigma W_1 \triangleq \Lambda\alpha.\,\lambda x_1{:}\alpha.\,\lambda x_2{:}\alpha.\,x_1$ or $M \cong_\sigma W_2 \triangleq \Lambda\alpha.\,\lambda x_1{:}\alpha.\,\lambda x_2{:}\alpha.\,x_2$

Proof By *extensionality*, it suffices to show that for either $i = 1$ or $i = 2$, for any closed type ρ and $V_1, V_2 : \rho$, we have $M\ \rho\ V_1\ V_2 \cong_\rho W_i\ \rho\ V_1\ V_2$, or just $M\ \rho\ V_1\ V_2 \cong_\sigma V_i$ (**1**).

Let ρ and $V_1, V_2 : \rho$ be fixed. Consider R equal to $\{(\mathsf{tt}, V_1), (\mathsf{ff}, V_2)\}$ in $\mathcal{R}(\mathsf{B}, \rho)$ and η be $\alpha \mapsto (\mathsf{B}, \rho, R)$. We have $(\mathsf{tt}, V_1) \in \mathcal{V}[\![\alpha]\!]_\eta$ since $R(\mathsf{tt}, V_1)$ and, similarly, $(\mathsf{ff}, V_2) \in \mathcal{V}[\![\alpha]\!]_\eta$.

We have $(M, M) \in \mathcal{E}[\![\sigma]\!]$ by parametricity. Hence, $(M\ \mathsf{B}\ \mathsf{tt}\ \mathsf{ff}, M\ \rho\ V_1\ V_2)$ is in $\mathcal{V}[\![\alpha]\!]_\eta$, which means that $(M\ \mathsf{B}\ \mathsf{tt}\ \mathsf{ff}, M\ \rho\ V_1\ V_2)$ reduces to a pair of values in R, which implies:

$$\bigvee \begin{cases} M\ \mathsf{B}\ \mathsf{tt}\ \mathsf{ff} \cong_{\mathsf{B}} \mathsf{tt} & \wedge\ \ M\ \rho\ V_1\ V_2 \cong_\rho V_1 \\ M\ \mathsf{B}\ \mathsf{tt}\ \mathsf{ff} \cong_{\mathsf{B}} \mathsf{ff} & \wedge\ \ M\ \rho\ V_1\ V_2 \cong_\rho V_2 \end{cases}$$

Since, $M\ \mathsf{B}\ \mathsf{tt}\ \mathsf{ff}$ is independent of ρ, V_1, and V_2, this actually shows (1).

Exercise Inhabitants of $\forall\alpha.\,\alpha \to \alpha$

Redo the proof that all inhabitants of of $\forall\alpha.\,\alpha \to \alpha$ are observationally equivalent to the identity, following the schema that we used for booleans.

Applications

Inhabitants of $\forall\alpha.\,(\alpha \to \alpha) \to \alpha \to \alpha$

?

Applications Inhabitants of $\forall\alpha.\,(\alpha \to \alpha) \to \alpha \to \alpha$

Fact Let *nat* be $\forall\alpha.\,(\alpha \to \alpha) \to \alpha \to \alpha$. If $M : \mathit{nat}$, then $M \simeq_{\mathit{nat}} N_n$ for some integer n, where $N_n \triangleq \Lambda\alpha.\,\lambda f\!:\!\alpha \to \alpha.\,\lambda x\!:\!\alpha.\,f^n\,x$.

?

Applications Inhabitants of $\forall\alpha.\,(\alpha \to \alpha) \to \alpha \to \alpha$

Fact Let *nat* be $\forall\alpha.\,(\alpha \to \alpha) \to \alpha \to \alpha$. If $M : \textit{nat}$, then $M \simeq_{\textit{nat}} N_n$ for some integer n, where $N_n \triangleq \Lambda\alpha.\,\lambda f{:}\alpha \to \alpha.\,\lambda x{:}\alpha.\,f^n\,x$.

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

Applications Inhabitants of $\forall\alpha.\,(\alpha \to \alpha) \to \alpha \to \alpha$

Fact Let *nat* be $\forall\alpha.\,(\alpha \to \alpha) \to \alpha \to \alpha$. If $M : \mathit{nat}$, then $M \cong_{\mathit{nat}} N_n$ for some integer n, where $N_n \triangleq \Lambda\alpha.\,\lambda f{:}\alpha \to \alpha.\,\lambda x{:}\alpha.\,f^n\,x$.

Proof

?

Applications — Inhabitants of $\forall\alpha.\,(\alpha \to \alpha) \to \alpha \to \alpha$

Fact Let *nat* be $\forall\alpha.\,(\alpha \to \alpha) \to \alpha \to \alpha$. If $M : \mathit{nat}$, then $M \cong_{\mathit{nat}} N_n$ for some integer n, where $N_n \triangleq \Lambda\alpha.\,\lambda f{:}\alpha \to \alpha.\,\lambda x{:}\alpha.\,f^n\,x$.

Proof By *extensionality*, it suffices to show that there exists n such that for any closed type ρ and closed values $V_1 : \rho \to \rho$ and $V_2 : \rho$, we have $M\ \rho\,V_1\ V_2 \cong_\rho N_n\ \rho\,V_1\ V_2$, or, by closure by inverse reduction and replacing observational by logical equivalence, $M\ \rho\,V_1\ V_2 \sim_\rho V_1^n\ V_2$ $(\mathbf{1})$, since $N_n\ \rho\,V_1\ V_2$ reduces to $V_1^n\ V_2$.

?

Applications Inhabitants of $\forall\alpha.\,(\alpha\to\alpha)\to\alpha\to\alpha$

Fact Let *nat* be $\forall\alpha.\,(\alpha\to\alpha)\to\alpha\to\alpha$. If $M : \mathit{nat}$, then $M \cong_{\mathit{nat}} N_n$ for some integer n, where $N_n \triangleq \Lambda\alpha.\,\lambda f{:}\alpha\to\alpha.\,\lambda x{:}\alpha.\,f^n\,x$.

Proof By *extensionality*, it suffices to show that there exists n such that for any closed type ρ and closed values $V_1 : \rho\to\rho$ and $V_2 : \rho$, we have $M\;\rho\,V_1\;V_2 \cong_\rho N_n\;\rho\,V_1\;V_2$, or, by closure by inverse reduction and replacing observational by logical equivalence, $M\;\rho\,V_1\;V_2 \sim_\rho V_1^n\;V_2$ $(\mathbf{1})$, since $N_n\;\rho\,V_1\;V_2$ reduces to $V_1^n\;V_2$. Let ρ and $V_1 : \rho\to\rho$ and $V_2 : \rho$ be fixed.

Let Z be $N_0\;\mathit{nat}$ and S be $N_1\;\mathit{nat}$. Let R in $\mathcal{R}(\mathit{nat},\rho)$ be $\{(\mathsf{S}^k\;\mathsf{Z}, V_1^k\;V_2) \mid k\in\mathbb{N}\}$ and η be $\alpha\mapsto(\mathit{nat},\rho,R)$.

We have $(\mathsf{Z}, V_2)\in\mathcal{V}[\![\alpha]\!]_\eta$.
We also have $(\mathsf{S}, V_1)\in\mathcal{V}[\![\alpha\to\alpha]\!]_\eta$.

Applications — Inhabitants of $\forall\alpha.\,(\alpha\to\alpha)\to\alpha\to\alpha$

Fact Let *nat* be $\forall\alpha.\,(\alpha\to\alpha)\to\alpha\to\alpha$. If $M : \mathit{nat}$, then $M \cong_{\mathit{nat}} N_n$ for some integer n, where $N_n \triangleq \Lambda\alpha.\,\lambda f{:}\alpha\to\alpha.\,\lambda x{:}\alpha.\,f^n\;x$.

Proof By *extensionality*, it suffices to show that there exists n such that for any closed type ρ and closed values $V_1 : \rho\to\rho$ and $V_2 : \rho$, we have $M\;\rho\;V_1\;V_2 \cong_\rho N_n\;\rho\;V_1\;V_2$, or, by closure by inverse reduction and replacing observational by logical equivalence, $M\;\rho\;V_1\;V_2 \sim_\rho V_1^n\;V_2$ $(\mathbf{1})$, since $N_n\;\rho\;V_1\;V_2$ reduces to $V_1^n\;V_2$. Let ρ and $V_1 : \rho\to\rho$ and $V_2 : \rho$ be fixed.

Let Z be N_0 *nat* and S be N_1 *nat*. Let R in $\mathcal{R}(\mathit{nat},\rho)$ be $\{(S^k\;Z, V_1^k\;V_2) \mid k\in\mathbb{N}\}$ and η be $\alpha\mapsto(\mathit{nat},\rho,R)$.

We have $(Z, V_2)\in\mathcal{V}[\![\alpha]\!]_\eta$.
We also have $(S, V_1)\in\mathcal{V}[\![\alpha\to\alpha]\!]_\eta$. (A key to the proof.)

Indeed,

?

Applications — Inhabitants of $\forall\alpha.\,(\alpha \to \alpha) \to \alpha \to \alpha$

Fact Let *nat* be $\forall\alpha.\,(\alpha \to \alpha) \to \alpha \to \alpha$. If $M : \mathit{nat}$, then $M \cong_{\mathit{nat}} N_n$ for some integer n, where $N_n \triangleq \Lambda\alpha.\,\lambda f{:}\alpha \to \alpha.\,\lambda x{:}\alpha.\,f^n\,x$.

Proof By *extensionality*, it suffices to show that there exists n such that for any closed type ρ and closed values $V_1 : \rho \to \rho$ and $V_2 : \rho$, we have $M\;\rho\;V_1\;V_2 \cong_\rho N_n\;\rho\;V_1\;V_2$, or, by closure by inverse reduction and replacing observational by logical equivalence, $M\;\rho\;V_1\;V_2 \sim_\rho V_1^n\;V_2$ $(\mathbf{1})$, since $N_n\;\rho\;V_1\;V_2$ reduces to $V_1^n\;V_2$. Let ρ and $V_1 : \rho \to \rho$ and $V_2 : \rho$ be fixed.

Let Z be $N_0\;\mathit{nat}$ and S be $N_1\;\mathit{nat}$. Let R in $\mathcal{R}(\mathit{nat}, \rho)$ be $\{(S^k\;Z, V_1^k\;V_2) \mid k \in \mathbb{N}\}$ and η be $\alpha \mapsto (\mathit{nat}, \rho, R)$.

We have $(Z, V_2) \in \mathcal{V}[\![\alpha]\!]_\eta$.
We also have $(S, V_1) \in \mathcal{V}[\![\alpha \to \alpha]\!]_\eta$. (A key to the proof.)

Indeed, assume (W_1, W_2) in $\mathcal{V}[\![\alpha]\!]_\eta$. There exists k such that $W_1 = S^k\;Z$ and $W_2 = V_1^k\;V_2$. Thus, $(S\,W_1, V_1\;W_2)$ equal to $(S^{k+1}\;Z, V_1^{k+1}\;V_2)$ is in $\mathcal{E}[\![\alpha]\!]_\eta$.

Applications Inhabitants of $\forall\alpha.\,(\alpha \to \alpha) \to \alpha \to \alpha$

Fact Let *nat* be $\forall\alpha.\,(\alpha \to \alpha) \to \alpha \to \alpha$. If $M : \mathit{nat}$, then $M \cong_{\mathit{nat}} N_n$ for some integer n, where $N_n \triangleq \Lambda\alpha.\,\lambda f{:}\alpha \to \alpha.\,\lambda x{:}\alpha.\,f^n\,x$.

Proof By *extensionality*, it suffices to show that there exists n such that for any closed type ρ and closed values $V_1 : \rho \to \rho$ and $V_2 : \rho$, we have $M\ \rho\ V_1\ V_2 \cong_\rho N_n\ \rho\ V_1\ V_2$, or, by closure by inverse reduction and replacing observational by logical equivalence, $M\ \rho\ V_1\ V_2 \sim_\rho V_1^n\ V_2$ $(\mathbf{1})$, since $N_n\ \rho\ V_1\ V_2$ reduces to $V_1^n\ V_2$. Let ρ and $V_1 : \rho \to \rho$ and $V_2 : \rho$ be fixed.

Let Z be $N_0\ \mathit{nat}$ and S be $N_1\ \mathit{nat}$. Let R in $\mathcal{R}(\mathit{nat}, \rho)$ be $\{(S^k\ Z, V_1^k\ V_2) \mid k \in \mathbb{N}\}$ and η be $\alpha \mapsto (\mathit{nat}, \rho, R)$.

We have $(Z, V_2) \in \mathcal{V}[\![\alpha]\!]_\eta$.
We also have $(S, V_1) \in \mathcal{V}[\![\alpha \to \alpha]\!]_\eta$. (A key to the proof.)

?

Applications Inhabitants of $\forall\alpha.(\alpha\to\alpha)\to\alpha\to\alpha$

Fact Let *nat* be $\forall\alpha.(\alpha\to\alpha)\to\alpha\to\alpha$. If $M : \mathit{nat}$, then $M \cong_{\mathit{nat}} N_n$ for some integer n, where $N_n \triangleq \Lambda\alpha.\,\lambda f{:}\alpha\to\alpha.\,\lambda x{:}\alpha.\,f^n\,x$.

Proof By *extensionality*, it suffices to show that there exists n such that for any closed type ρ and closed values $V_1 : \rho\to\rho$ and $V_2 : \rho$, we have $M\ \rho\ V_1\ V_2 \cong_\rho N_n\ \rho\ V_1\ V_2$, or, by closure by inverse reduction and replacing observational by logical equivalence, $M\ \rho\ V_1\ V_2 \sim_\rho V_1^n\ V_2$ $(\mathbf{1})$, since $N_n\ \rho\ V_1\ V_2$ reduces to $V_1^n\ V_2$. Let ρ and $V_1 : \rho\to\rho$ and $V_2 : \rho$ be fixed.

Let Z be $N_0\ \mathit{nat}$ and S be $N_1\ \mathit{nat}$. Let R in $\mathcal{R}(\mathit{nat},\rho)$ be $\{(\mathsf{S}^k\ \mathsf{Z}, V_1^k\ V_2) \mid k\in\mathbb{N}\}$ and η be $\alpha\mapsto(\mathit{nat},\rho,R)$.

We have $(\mathsf{Z}, V_2)\in\mathcal{V}[\![\alpha]\!]_\eta$.
We also have $(\mathsf{S}, V_1)\in\mathcal{V}[\![\alpha\to\alpha]\!]_\eta$. (A key to the proof.)

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

?

Applications

Inhabitants of $\forall\alpha.\,(\alpha \to \alpha) \to \alpha \to \alpha$

Fact Let *nat* be $\forall\alpha.\,(\alpha \to \alpha) \to \alpha \to \alpha$. If $M : \mathit{nat}$, then $M \cong_{\mathit{nat}} N_n$ for some integer n, where $N_n \triangleq \Lambda\alpha.\,\lambda f{:}\alpha \to \alpha.\,\lambda x{:}\alpha.\,f^n\,x$.

Proof By *extensionality*, it suffices to show that there exists n such that for any closed type ρ and closed values $V_1 : \rho \to \rho$ and $V_2 : \rho$, we have $M\;\rho\,V_1\;V_2 \cong_\rho N_n\;\rho\,V_1\;V_2$, or, by closure by inverse reduction and replacing observational by logical equivalence, $M\;\rho\,V_1\;V_2 \sim_\rho V_1^n\;V_2$ $(\mathbf{1})$, since $N_n\;\rho\,V_1\;V_2$ reduces to $V_1^n\;V_2$. Let ρ and $V_1 : \rho \to \rho$ and $V_2 : \rho$ be fixed.

Let Z be N_0 *nat* and S be N_1 *nat*. Let R in $\mathcal{R}(\mathit{nat}, \rho)$ be $\{(S^k\;Z, V_1^k\;V_2) \mid k \in \mathbb{N}\}$ and η be $\alpha \mapsto (\mathit{nat}, \rho, R)$.

We have $(Z, V_2) \in \mathcal{V}[\![\alpha]\!]_\eta$.
We also have $(S, V_1) \in \mathcal{V}[\![\alpha \to \alpha]\!]_\eta$. (A key to the proof.)

By parametricity, we have $M \sim_{\mathit{nat}} M$. Hence, $(M\;\mathit{nat}\;S\;Z, M\;\rho\,V_1\;V_2) \in \mathcal{E}[\![\alpha]\!]_\eta$.
Thus, there exists n such that $M\;\mathit{nat}\;S\;Z \cong_{\mathit{nat}} S^n\;Z$ and $M\;\rho\,V_1\;V_2 \cong_\rho V_1^n\;V_2$.

?

Applications Inhabitants of $\forall\alpha.\,(\alpha \to \alpha) \to \alpha \to \alpha$

Fact Let *nat* be $\forall\alpha.\,(\alpha \to \alpha) \to \alpha \to \alpha$. If $M : \textit{nat}$, then $M \cong_{\textit{nat}} N_n$ for some integer n, where $N_n \triangleq \Lambda\alpha.\,\lambda f\!:\!\alpha \to \alpha.\,\lambda x\!:\!\alpha.\,f^n\,x$.

Proof By *extensionality*, it suffices to show that there exists n such that for any closed type ρ and closed values $V_1 : \rho \to \rho$ and $V_2 : \rho$, we have $M\ \rho\,V_1\ V_2 \cong_\rho N_n\ \rho\,V_1\ V_2$, or, by closure by inverse reduction and replacing observational by logical equivalence, $M\ \rho\,V_1\ V_2 \sim_\rho V_1^n\ V_2$ $(\mathbf{1})$, since $N_n\ \rho\,V_1\ V_2$ reduces to $V_1^n\ V_2$. Let ρ and $V_1 : \rho \to \rho$ and $V_2 : \rho$ be fixed.

Let Z be $N_0\ \textit{nat}$ and S be $N_1\ \textit{nat}$. Let R in $\mathcal{R}(\textit{nat}, \rho)$ be $\{(S^k\ Z, V_1^k\ V_2) \mid k \in \mathbb{N}\}$ and η be $\alpha \mapsto (\textit{nat}, \rho, R)$.

We have $(Z, V_2) \in \mathcal{V}[\![\alpha]\!]_\eta$.
We also have $(S, V_1) \in \mathcal{V}[\![\alpha \to \alpha]\!]_\eta$. (A key to the proof.)

By parametricity, we have $M \sim_{\textit{nat}} M$. Hence, $(M\ \textit{nat}\ S\ Z, M\ \rho\,V_1\ V_2) \in \mathcal{E}[\![\alpha]\!]_\eta$.
Thus, there exists n such that $M\ \textit{nat}\ S\ Z \cong_{\textit{nat}} S^n\ Z$ and $M\ \rho\,V_1\ V_2 \cong_\rho V_1^n\ V_2$.

Since, $M\ \textit{nat}\ S\ Z$ is independent of n, we may conclude (1), provided the $S^n\ Z$ are all in different observational equivalence classes (easy to check).

Applications *Inhabitants of* $\forall \alpha.\, \alpha \to (\tau \to \alpha \to \alpha) \to \alpha$

- Left as an exercise. . .

Applications $\forall\alpha.\,\alpha \to (\tau \to \alpha \to \alpha) \to \alpha$

Fact Let τ be closed and *list* be $\forall\alpha.\,\alpha \to (\tau \to \alpha \to \alpha) \to \alpha$. Let C be $\lambda H{:}\tau.\,\lambda T{:}\textit{list}\,.\,\Lambda\alpha.\,\lambda n{:}\alpha.\,\lambda c{:}\tau \to \alpha \to \alpha.\,c\;H\;(T\;\alpha\;n\;c)$ and N be $\Lambda\alpha.\,\lambda n{:}\alpha.\,\lambda c{:}\tau \to \alpha \to \alpha.\,n$. If $M : \textit{list}$, then $M \cong_{\textit{list}} N_n$ for some N_n in $\mathcal{L}_n$ where $\mathcal{L}_k$ is defined inductively by

$$\mathcal{L}_0 \triangleq \{N\} \text{ and } \mathcal{L}_{k+1} \triangleq \{C\,W_k\;N_k \mid W_k \in \mathrm{Val}(\tau) \wedge N_k \in \mathcal{L}_k\}$$

Proof

?

Applications $\forall\alpha.\,\alpha \to (\tau \to \alpha \to \alpha) \to \alpha$

Fact Let τ be closed and *list* be $\forall\alpha.\,\alpha \to (\tau \to \alpha \to \alpha) \to \alpha$. Let C be $\lambda H{:}\tau.\,\lambda T{:}\mathit{list}\,.\,\Lambda\alpha.\,\lambda n{:}\alpha.\,\lambda c{:}\tau \to \alpha \to \alpha.\,c\,H\,(T\;\alpha\;n\;c)$ and N be $\Lambda\alpha.\,\lambda n{:}\alpha.\,\lambda c{:}\tau \to \alpha \to \alpha.\,n$. If $M : \mathit{list}$, then $M \cong_{\mathit{list}} N_n$ for some N_n in $\mathcal{L}_n$ where $\mathcal{L}_k$ is defined inductively by

$$\mathcal{L}_0 \triangleq \{N\} \text{ and } \mathcal{L}_{k+1} \triangleq \{C\,W_k\;N_k \mid W_k \in \mathrm{Val}(\tau) \wedge N_k \in \mathcal{L}_k\}$$

Proof By *extensionality*, it suffices to show that there exists n and $N_n \in \mathcal{L}_n$ such that for any closed type ρ and closed values $V_1 : \tau \to \rho \to \rho$ and $V_2 : \rho$, we have $M\;\rho\,V_1\;V_2 \sim_\rho N_n\;\rho\,V_1\;V_2$, or, by closure by inverse reduction and replacing observational by logical equivalence, $C\;W_n\;(\ldots(C\;W_1\;N)\ldots)$ (**1**), since $N_n\;\rho\,V_1\;V_2$ reduces to $C\;W_n\;(\ldots(C\;W_1\;N)\ldots)$ where all W_k are in $\mathrm{Val}(\tau)$.

Let ρ and $V_1 : \alpha \to \rho \to \rho$ and $V_2 : \rho$ be fixed.

Let R in $\mathcal{R}(\mathit{list}, \rho)$ be defined inductively as $\bigcup R_n$ where R_{k+1} is $\{\Downarrow (C\,G\;T, V_2\;H\;U) \mid (G, H) \in \mathcal{V}[\![\tau]\!]_\eta \wedge (T, U) \in R_k\}$ and R_0 is $\{(N, V_1)\}$.

We have $(N, V_1) \in R_0 \subseteq \mathcal{V}[\![\alpha]\!]_\eta$.
We also have $(C,\;V_2) \in \mathcal{V}[\![\tau \to \alpha \to \alpha]\!]_\eta$.

Applications $\forall\alpha.\,\alpha \to (\tau \to \alpha \to \alpha) \to \alpha$ ▷

Fact Let τ be closed and *list* be $\forall\alpha.\,\alpha \to (\tau \to \alpha \to \alpha) \to \alpha$. If $M : \textit{list}$, then $M \cong_{\textit{list}} N_n$ for some N_n in $\mathcal{L}_n$ where $\mathcal{L}_k$ is defined inductively by $\mathcal{L}_0 \triangleq \{\mathsf{N}\}$ and $\mathcal{L}_{k+1} \triangleq \{\mathsf{C}\, W_k\, N_k \mid W_k \in \mathsf{Val}(\tau) \wedge N_k \in \mathcal{L}_k\}$.

Proof By *extensionality*, it suffices to show that there exists n and $N_n \in \mathcal{L}_n$ such that for any closed type ρ and closed values $V_1 : \tau \to \rho \to \rho$ and $V_2 : \rho$, we have $\mathsf{C}\ W_n\ (\ldots(\mathsf{C}\ W_1\ \mathsf{N})\ldots)\ (\mathbf{1})$.

Let ρ and $V_1 : \alpha \to \rho \to \rho$ and $V_2 : \rho$ be fixed. Let R in $\mathcal{R}(\textit{list}, \rho)$ be defined inductively as $\bigcup R_n$ where R_{k+1} is $\{\Downarrow (\mathsf{C}\, G\, T, V_2\ H\ U) \mid (G, H) \in \mathcal{V}[\![\tau]\!]_\eta \wedge (T, U) \in R_k\}$ and R_0 is $\{(\mathsf{N}, V_1)\}$.

We have $(\mathsf{N}, V_1) \in R_0 \subseteq \mathcal{V}[\![\alpha]\!]_\eta$.

We also have $(\mathsf{C},\ V_2) \in \mathcal{V}[\![\tau \to \alpha \to \alpha]\!]_\eta$. (A key to the proof)

Indeed,

?

Applications $\forall \alpha. \alpha \to (\tau \to \alpha \to \alpha) \to \alpha$

Fact Let τ be closed and *list* be $\forall \alpha. \alpha \to (\tau \to \alpha \to \alpha) \to \alpha$. If $M : \textit{list}$, then $M \cong_{\textit{list}} N_n$ for some N_n in $\mathcal{L}_n$ where $\mathcal{L}_k$ is defined inductively by $\mathcal{L}_0 \triangleq \{N\}$ and $\mathcal{L}_{k+1} \triangleq \{C\, W_k\, N_k \mid W_k \in \mathsf{Val}(\tau) \wedge N_k \in \mathcal{L}_k\}$.

Proof By *extensionality*, it suffices to show that there exists n and $N_n \in \mathcal{L}_n$ such that for any closed type ρ and closed values $V_1 : \tau \to \rho \to \rho$ and $V_2 : \rho$, we have $C\ W_n\ (\ldots (C\ W_1\ N) \ldots)\ (\mathbf{1})$.

Let ρ and $V_1 : \alpha \to \rho \to \rho$ and $V_2 : \rho$ be fixed. Let R in $\mathcal{R}(\textit{list}, \rho)$ be defined inductively as $\bigcup R_n$ where R_{k+1} is $\{\Downarrow (C\, G\, T, V_2\ H\ U) \mid (G, H) \in \mathcal{V}[\![\tau]\!]_\eta \wedge (T, U) \in R_k\}$ and R_0 is $\{(N, V_1)\}$.

We have $(N, V_1) \in R_0 \subseteq \mathcal{V}[\![\alpha]\!]_\eta$.

We also have $(C,\ V_2) \in \mathcal{V}[\![\tau \to \alpha \to \alpha]\!]_\eta$. (A key to the proof)

Indeed, assume (G, H) in $\mathcal{V}[\![\tau]\!]_\eta$ and (T, U) in $\mathcal{V}[\![\alpha]\!]_\eta$, *i.e.* in R_k for some k. Then, $\Downarrow (C\, G\, T, V_2\ H\ U)$ is in $R^{k+1} \subseteq \mathcal{V}[\![\alpha]\!]_\eta$. Hence, $(C\, G\, T, V_2\ H\ U) \in \mathcal{E}[\![\alpha]\!]_\eta$, as expected.

Applications $\forall\alpha.\,\alpha \to (\tau \to \alpha \to \alpha) \to \alpha$ ▷

Fact Let τ be closed and *list* be $\forall\alpha.\,\alpha \to (\tau \to \alpha \to \alpha) \to \alpha$. If $M : \mathit{list}$, then $M \cong_{\mathit{list}} N_n$ for some N_n in $\mathcal{L}_n$ where $\mathcal{L}_k$ is defined inductively by $\mathcal{L}_0 \triangleq \{N\}$ and $\mathcal{L}_{k+1} \triangleq \{C\,W_k\,N_k \mid W_k \in \mathrm{Val}(\tau) \wedge N_k \in \mathcal{L}_k\}$.

Proof By *extensionality*, it suffices to show that there exists n and $N_n \in \mathcal{L}_n$ such that for any closed type ρ and closed values $V_1 : \tau \to \rho \to \rho$ and $V_2 : \rho$, we have $C\ W_n\ (\ldots(C\ W_1\ N)\ldots)\ (\mathbf{1})$.

Let ρ and $V_1 : \alpha \to \rho \to \rho$ and $V_2 : \rho$ be fixed. Let R in $\mathcal{R}(\mathit{list}, \rho)$ be defined inductively as $\bigcup R_n$ where R_{k+1} is $\{\Downarrow (C\,G\,T, V_2\ H\ U) \mid (G,H) \in \mathcal{V}[\![\tau]\!]_\eta \wedge (T,U) \in R_k\}$ and R_0 is $\{(N, V_1)\}$.

We have $(N, V_1) \in R_0 \subseteq \mathcal{V}[\![\alpha]\!]_\eta$. We also have $(C,\ V_2) \in \mathcal{V}[\![\tau \to \alpha \to \alpha]\!]_\eta$.

?

Applications $\forall\alpha.\,\alpha \to (\tau \to \alpha \to \alpha) \to \alpha$ ▷

Fact Let τ be closed and *list* be $\forall\alpha.\,\alpha \to (\tau \to \alpha \to \alpha) \to \alpha$. If $M : \textit{list}$, then $M \cong_{\textit{list}} N_n$ for some N_n in $\mathcal{L}_n$ where $\mathcal{L}_k$ is defined inductively by $\mathcal{L}_0 \triangleq \{N\}$ and $\mathcal{L}_{k+1} \triangleq \{C\,W_k\,N_k \mid W_k \in \mathrm{Val}(\tau) \wedge N_k \in \mathcal{L}_k\}$.

Proof By *extensionality*, it suffices to show that there exists n and $N_n \in \mathcal{L}_n$ such that for any closed type ρ and closed values $V_1 : \tau \to \rho \to \rho$ and $V_2 : \rho$, we have $C\ W_n\ (\ldots(C\ W_1\ N)\ldots)\ (\mathbf{1})$.

Let ρ and $V_1 : \alpha \to \rho \to \rho$ and $V_2 : \rho$ be fixed. Let R in $\mathcal{R}(\textit{list}, \rho)$ be defined inductively as $\bigcup R_n$ where R_{k+1} is
$\{\Downarrow (C\,G\,T, V_2\ H\ U) \mid (G,H) \in \mathcal{V}[\![\tau]\!]_\eta \wedge (T,U) \in R_k\}$ and R_0 is $\{(N, V_1)\}$.

We have $(N, V_1) \in R_0 \subseteq \mathcal{V}[\![\alpha]\!]_\eta$. We also have $(C,\ V_2) \in \mathcal{V}[\![\tau \to \alpha \to \alpha]\!]_\eta$.

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

?

Applications $\forall\alpha.\,\alpha \to (\tau \to \alpha \to \alpha) \to \alpha$

Fact Let τ be closed and *list* be $\forall\alpha.\,\alpha \to (\tau \to \alpha \to \alpha) \to \alpha$. If $M : \textit{list}$, then $M \cong_{\textit{list}} N_n$ for some N_n in $\mathcal{L}_n$ where $\mathcal{L}_k$ is defined inductively by $\mathcal{L}_0 \triangleq \{\mathsf{N}\}$ and $\mathcal{L}_{k+1} \triangleq \{\mathsf{C}\, W_k\, N_k \mid W_k \in \mathsf{Val}(\tau) \wedge N_k \in \mathcal{L}_k\}$.

Proof By *extensionality*, it suffices to show that there exists n and $N_n \in \mathcal{L}_n$ such that for any closed type ρ and closed values $V_1 : \tau \to \rho \to \rho$ and $V_2 : \rho$, we have $\mathsf{C}\ W_n\ (\ldots(\mathsf{C}\ W_1\ \mathsf{N})\ldots)\ (\mathbf{1})$.

Let ρ and $V_1 : \alpha \to \rho \to \rho$ and $V_2 : \rho$ be fixed. Let R in $\mathcal{R}(\textit{list}, \rho)$ be defined inductively as $\bigcup R_n$ where R_{k+1} is $\{\Downarrow (\mathsf{C}\, G\, T, V_2\ H\ U) \mid (G, H) \in \mathcal{V}[\![\tau]\!]_\eta \wedge (T, U) \in R_k\}$ and R_0 is $\{(\mathsf{N}, V_1)\}$.

We have $(\mathsf{N}, V_1) \in R_0 \subseteq \mathcal{V}[\![\alpha]\!]_\eta$. We also have $(\mathsf{C},\ V_2) \in \mathcal{V}[\![\tau \to \alpha \to \alpha]\!]_\eta$.

By parametricity, we have $M \sim_{\textit{list}} M$. Hence, $(M\ \textit{list}\ \mathsf{C}\ \mathsf{N}, M\ \rho\ V_1\ V_2) \in \mathcal{E}[\![\alpha]\!]_\eta$. Thus, there exists n such that $M\ \textit{list}\ \mathsf{C}\ \mathsf{N} \cong_{\textit{list}} \mathsf{C}\ W_n\ (\ldots(\mathsf{C}\ W_1\ \mathsf{N})\ldots)$ and $M\ \rho\ V_1\ V_2 \cong_\rho V_2\ W_n\ (\ldots(V_2\ W_1\ V_1)\ldots)$.

?

Applications $\forall \alpha. \alpha \rightarrow (\tau \rightarrow \alpha \rightarrow \alpha) \rightarrow \alpha$

Fact Let τ be closed and *list* be $\forall \alpha. \alpha \rightarrow (\tau \rightarrow \alpha \rightarrow \alpha) \rightarrow \alpha$. If $M : \textit{list}$, then $M \simeq_{\textit{list}} N_n$ for some N_n in $\mathcal{L}_n$ where $\mathcal{L}_k$ is defined inductively by $\mathcal{L}_0 \triangleq \{\mathsf{N}\}$ and $\mathcal{L}_{k+1} \triangleq \{\mathsf{C}\, W_k\, N_k \mid W_k \in \mathsf{Val}(\tau) \wedge N_k \in \mathcal{L}_k\}$.

Proof By *extensionality*, it suffices to show that there exists n and $N_n \in \mathcal{L}_n$ such that for any closed type ρ and closed values $V_1 : \tau \rightarrow \rho \rightarrow \rho$ and $V_2 : \rho$, we have $\mathsf{C}\ W_n\ (\ldots (\mathsf{C}\ W_1\ \mathsf{N}) \ldots)$ $(\mathbf{1})$.

Let ρ and $V_1 : \alpha \rightarrow \rho \rightarrow \rho$ and $V_2 : \rho$ be fixed. Let R in $\mathcal{R}(\textit{list}, \rho)$ be defined inductively as $\bigcup R_n$ where R_{k+1} is $\{\Downarrow (\mathsf{C}\, G\, T, V_2\ H\ U) \mid (G, H) \in \mathcal{V}[\![\tau]\!]_\eta \wedge (T, U) \in R_k\}$ and R_0 is $\{(\mathsf{N}, V_1)\}$.

We have $(\mathsf{N}, V_1) \in R_0 \subseteq \mathcal{V}[\![\alpha]\!]_\eta$. We also have $(\mathsf{C},\ V_2) \in \mathcal{V}[\![\tau \rightarrow \alpha \rightarrow \alpha]\!]_\eta$.

By parametricity, we have $M \sim_{\textit{list}} M$. Hence, $(M\ \textit{list}\ \mathsf{C}\, \mathsf{N}, M\ \rho\, V_1\ V_2) \in \mathcal{E}[\![\alpha]\!]_\eta$. Thus, there exists n such that $M\ \textit{list}\ \mathsf{C}\, \mathsf{N} \simeq_{\textit{list}} \mathsf{C}\ W_n\ (\ldots (\mathsf{C}\ W_1\ \mathsf{N}) \ldots)$ and $M\ \rho\, V_1\ V_2 \simeq_\rho V_2\, W_n\ (\ldots (V_2\, W_1\ V_1) \ldots)$.

Since, $M\ \textit{list}\ \mathsf{C}\, \mathsf{N}$ is independent of n and $(W_k)_{k \in 1..n}$, we may conclude (1). (This uses that $\mathcal{R}_k$ are all in different observational equivalence classes, which is easy to check, as a length function would return different integers.)

Contents

Encodable features Natural numbers

We have shown that all expressions of type *nat* behave as natural numbers. Hence, natural numbers are definable.

Still, we could also provide a type *nat* of natural numbers as primitive.

Then, we may extend

- behavioral equivalence: if $M_1 : \mathit{nat}$ and $M_2 : \mathit{nat}$, we have $M_1 \simeq_{\mathit{nat}} M_2$ iff there exists $n : \mathit{nat}$ such that $M_1 \Downarrow n$ and $M_2 \Downarrow n$.
- logical equivalence: $\mathcal{V}[\![\mathit{nat}]\!] \triangleq \{(n, n) \mid n \in \mathbb{N}\}$

All properties are preserved.

Encodable features — Products

Given closed types τ_1 and τ_2, we defined

$$\begin{aligned} \tau_1 \times \tau_2 &\triangleq \forall \alpha. (\tau_1 \to \tau_2 \to \alpha) \to \alpha \\ (M_1, M_2) &\triangleq \Lambda \alpha. \lambda x{:}\tau_1 \to \tau_2 \to \alpha.\, x\; M_1\; M_2 \\ M.i &\triangleq M\; (\lambda x_1{:}\tau_1. \lambda x_2{:}\tau_2. x_i) \end{aligned}$$

Facts

If $M : \tau_1 \times \tau_2$, then $M \cong_{\tau_1 \times \tau_2} (M_1, M_2)$ for some $M_1 : \tau_1$ and $M_2 : \tau_2$.

If $M : \tau_1 \times \tau_2$ and $M.1 \cong_{\tau_1} M_1$ and $M.2 \cong_{\tau_2} M_2$, then $M \cong_{\tau_1 \times \tau_2} (M_1, M_2)$

Primitive pairs

We may instead extend the language with *primitive* pairs. Then,

$$\mathcal{V}[\![\tau \times \sigma]\!]_\eta \triangleq \{((V_1, W_1), (V_2, W_2)) \mid (V_1, V_2) \in \mathcal{V}[\![\tau]\!]_\eta \wedge (W_1, W_2) \in \mathcal{V}[\![\sigma]\!]_\eta\}$$

Sums

We define:

$$\begin{array}{rcl} \mathcal{V}[\![\tau + \sigma]\!]_\eta & = & \{(\mathit{inj}_1\ V_1, \mathit{inj}_1\ V_2) \mid (V_1, V_2) \in \mathcal{V}[\![\tau]\!]_\eta\} \ \cup \\ & & \{(\mathit{inj}_2\ W_1, \mathit{inj}_2\ W_2) \mid (W_1, W_2) \in \mathcal{V}[\![\sigma]\!]_\eta\} \end{array}$$

Notice that sums, as all datatypes, can also be encoded in System F.

Primitive Lists

We *recursively*[1] define

$$\begin{aligned}
\mathcal{V}[\![\mathit{list}\ \tau]\!]_\eta &\triangleq \textstyle\bigcup_k \mathcal{W}^k_\eta \\
\text{where} \quad \mathcal{W}^0_\eta &= \{(\mathit{Nil}, \mathit{Nil})\} \\
\mathcal{W}^{k+1}_\eta &= \{(\mathit{Cons}\ H_1\ T_1, \mathit{Cons}\ H_2\ T_2) \\
&\qquad \mid\ (H_1, H_2) \in \mathcal{V}[\![\tau]\!]_\eta \ \wedge\ (T_1, T_2) \in \mathcal{W}^k_\eta\}
\end{aligned}$$

[1]This definition is well-founded.

Primitive Lists

We *recursively*[1] define

$$\begin{aligned}
\mathcal{V}[\![\mathit{list}\,\tau]\!]_\eta &\triangleq \textstyle\bigcup_k \mathcal{W}^{\,k}_\eta \\
\text{where}\quad \mathcal{W}^{\,0}_\eta &= \{(\mathit{Nil}, \mathit{Nil})\} \\
\mathcal{W}^{\,k+1}_\eta &= \{(\mathit{Cons}\; H_1\; T_1, \mathit{Cons}\; H_2\; T_2) \\
&\qquad \mid\ (H_1, H_2) \in \mathcal{V}[\![\tau]\!]_\eta \ \wedge\ (T_1, T_2) \in \mathcal{W}^{\,k}_\eta\}
\end{aligned}$$

Assume that $(\alpha \mapsto \rho_1, \rho_2, R) \in \eta$ where R in $\mathcal{R}(\rho_1, \rho_2)$ is the graph $\langle g \rangle$ of a function g, *i.e.* equal to $\{(V_1, V_2) \mid g\, V_1 \Downarrow V_2\}$. Then, we have:

$$\mathcal{V}[\![\mathit{list}\,\alpha]\!]_\eta(W_1, W_2)$$

$$\iff \quad \exists k, \bigvee \begin{cases} W_1 = \mathit{Nil} \wedge W_2 = \mathit{Nil} \\ W_1 = \mathit{Cons}\; H_1\; T_1 \wedge W_2 = \mathit{Cons}\; H_2\; T_2 \wedge g\; H_1 \Downarrow H_2 \\ \qquad\quad \wedge\, (T_1, T_2) \in \mathcal{W}^{\,k}_\eta \end{cases}$$

[1] This definition is well-founded.

Primitive Lists

We *recursively*[1] define

$$\begin{aligned}
\mathcal{V}[\![\mathit{list}\,\tau]\!]_\eta &\triangleq \textstyle\bigcup_k \mathcal{W}_\eta^k \\
\text{where}\quad \mathcal{W}_\eta^0 &= \{(\mathit{Nil}, \mathit{Nil})\} \\
\mathcal{W}_\eta^{k+1} &= \{(\mathit{Cons}\, H_1\, T_1, \mathit{Cons}\, H_2\, T_2) \\
&\qquad \mid\ (H_1, H_2) \in \mathcal{V}[\![\tau]\!]_\eta \ \wedge\ (T_1, T_2) \in \mathcal{W}_\eta^k\}
\end{aligned}$$

Assume that $(\alpha \mapsto \rho_1, \rho_2, R) \in \eta$ where R in $\mathcal{R}(\rho_1, \rho_2)$ is the graph $\langle g \rangle$ of a function g, *i.e.* equal to $\{(V_1, V_2) \mid g\, V_1 \Downarrow V_2\}$. Then, we have:

$$\mathcal{V}[\![\mathit{list}\,\alpha]\!]_\eta(W_1, W_2)$$

$$\iff \quad \exists k, \bigvee \begin{cases} W_1 = \mathit{Nil} \wedge W_2 = \mathit{Nil} \\ W_1 = \mathit{Cons}\, H_1\, T_1 \wedge W_2 \Downarrow \mathit{Cons}\,(g\, H_1)\, T_2 \\ \qquad \wedge (T_1, T_2) \in \mathcal{W}_\eta^k \end{cases}$$

[1] This definition is well-founded.

Primitive Lists

We *recursively*[1] define

$$\begin{array}{rcl} \mathcal{V}[\![\textit{list}\,\tau]\!]_\eta & \triangleq & \bigcup_k \mathcal{W}^k_\eta \\ \text{where} \quad \mathcal{W}^0_\eta & = & \{(\textit{Nil}, \textit{Nil})\} \\ \mathcal{W}^{k+1}_\eta & = & \{(\textit{Cons}\, H_1\, T_1, \textit{Cons}\, H_2\, T_2) \\ & & \quad \mid\ (H_1, H_2) \in \mathcal{V}[\![\tau]\!]_\eta \ \wedge\ (T_1, T_2) \in \mathcal{W}^k_\eta\} \end{array}$$

Assume that $(\alpha \mapsto \rho_1, \rho_2, R) \in \eta$ where R in $\mathcal{R}(\rho_1, \rho_2)$ is the graph $\langle g \rangle$ of a function g, *i.e.* equal to $\{(V_1, V_2) \mid g\, V_1 \Downarrow V_2\}$. Then, we have:

$$\begin{array}{l} \mathcal{V}[\![\textit{list}\,\alpha]\!]_\eta(W_1, W_2) \\ \iff \quad \exists k, \bigvee \left\{ \begin{array}{l} W_1 = \textit{Nil} \wedge W_2 = \textit{Nil} \\ W_1 = \textit{Cons}\, H_1\, T_1 \wedge W_2 \Downarrow \textit{Cons}\,(g\, H_1)\, T_2 \\ \qquad \wedge\, (T_1, T_2) \in \mathcal{W}^k_\eta \end{array} \right. \\ \iff \quad \textit{map}\, \rho_1\, \rho_2\, g\ W_1 \Downarrow W_2 \end{array}$$

[1] This definition is well-founded.

Primitive Lists

We *recursively*[1] define

$$\begin{aligned}\mathcal{V}[\![\textit{list}\,\tau]\!]_\eta &\triangleq \textstyle\bigcup_k \mathcal{W}_\eta^{k} \\ \text{where}\quad \mathcal{W}_\eta^{0} &= \{(\textit{Nil}, \textit{Nil})\} \\ \mathcal{W}_\eta^{k+1} &= \{(\textit{Cons}\, H_1\, T_1, \textit{Cons}\, H_2\, T_2) \\ &\qquad \mid\ (H_1, H_2) \in \mathcal{V}[\![\tau]\!]_\eta \ \wedge\ (T_1, T_2) \in \mathcal{W}_\eta^{k}\}\end{aligned}$$

Assume that $(\alpha \mapsto \rho_1, \rho_2, R) \in \eta$ where R in $\mathcal{R}(\rho_1, \rho_2)$ is the graph $\langle g \rangle$ of a function g, *i.e.* equal to $\{(V_1, V_2) \mid g\, V_1 \Downarrow V_2\}$. Then, we have:

$$\mathcal{V}[\![\textit{list}\,\alpha]\!]_\eta = \langle \textit{map}\, \rho_1\, \rho_2\, g \rangle$$

[1]This definition is well-founded.

Applications $\mathit{sort} : \forall \alpha. (\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\, \alpha$

Fact: Assume *sort* : $\forall \alpha. (\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\, \alpha \to \mathit{list}\, \alpha$ $(\mathbf{1})$. Then

$$(\forall x, y,\ \mathit{cmp}_2\ (f\ x)\ (f\ y) = \mathit{cmp}_1\ x\ y) \Longrightarrow \forall \ell,\ \mathit{sort}\ \mathit{cmp}_2\ (\mathit{map}\ f\ \ell) = \mathit{map}\ f\ (\mathit{sort}\ \mathit{cmp}_1\ \ell)$$

Applications $\mathit{sort} : \forall \alpha.\, (\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\,\alpha$

Fact: Assume $\mathit{sort} : \forall \alpha.\, (\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\,\alpha \to \mathit{list}\,\alpha$ $(\mathbf{1})$. Then

$$(\forall x, y,\;\; \mathit{cmp}_2\; (f\; x)\; (f\; y) = \mathit{cmp}_1\;\; x\; y) \implies \forall \ell,\;\; \mathit{sort}\;\; \mathit{cmp}_2\; (\mathit{map}\; f\; \ell) = \mathit{map}\; f\; (\mathit{sort}\; \mathit{cmp}_1\; \ell)$$

Applications $\mathit{sort} : \forall\alpha.\,(\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\,\alpha$

Fact: Assume *sort* : $\forall\alpha.\,(\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\,\alpha \to \mathit{list}\,\alpha$ $(\mathbf{1})$. Then

$$(\forall x, y,\ \ \mathit{cmp}_2\ (f\ x)\ (f\ y) \cong \mathit{cmp}_1\ \ x\ y) \implies \forall \ell,\ \ \mathit{sort}\ \ \mathit{cmp}_2\ (\mathit{map}\ f\ \ell) \cong \mathit{map}\ f\ (\mathit{sort}\ \mathit{cmp}_1\ \ell)$$

Applications $\mathit{sort} : \forall\alpha.\,(\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\;\alpha$

Proof: Assume $\forall x, y,\;\; \mathit{cp}\,(f\;x)\,(f\;y) \cong \mathit{cp}\;\,x\;y)$ (**H**).

We have $\mathit{sort} \sim_\sigma \mathit{sort}$ where σ is $\forall\alpha.\,(\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\;\alpha \to \mathit{list}\;\alpha$.

Thus, for all ρ_1, ρ_2, and relations R in $\mathcal{R}(\rho_1, \rho_2)$,

$$\forall(\mathit{cp}_1, \mathit{cp}_2) \in \mathcal{V}[\![\alpha \to \alpha \to \mathsf{B}]\!]_\eta, \tag{1}$$

$$\forall(V_1, V_2) \in \mathcal{V}[\![\mathit{list}\;\alpha]\!]_\eta,\;\; (\mathit{sort}\;\rho_1\;\mathit{cp}_1\;\;V_1, \mathit{sort}\;\rho_2\;\mathit{cp}_2\;\;V_2) \in \mathcal{E}[\![\mathit{list}\;\alpha]\!]_\eta) \tag{2}$$

where η is $\alpha \mapsto (\rho_1, \rho_2, R)$.

Applications $\mathit{sort} : \forall \alpha. (\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\ \alpha$

Proof: Assume $\forall x, y,\ \mathit{cp}\ (f\ x)\ (f\ y) \cong \mathit{cp}\ x\ y)$ (**H**).

We have $\mathit{sort} \sim_\sigma \mathit{sort}$ where σ is $\forall \alpha. (\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\ \alpha \to \mathit{list}\ \alpha$.

Thus, for all ρ_1, ρ_2, and relations R in $\mathcal{R}(\rho_1, \rho_2)$,

$$\forall (\mathit{cp}_1, \mathit{cp}_2) \in \mathcal{V}[\![\alpha \to \alpha \to \mathsf{B}]\!]_\eta, \qquad \textbf{(1)}$$

$$\forall (V_1, V_2) \in \mathcal{V}[\![\mathit{list}\ \alpha]\!]_\eta,\ (\mathit{sort}\ \rho_1\ \mathit{cp}_1\ \ V_1, \mathit{sort}\ \rho_2\ \mathit{cp}_2\ \ V_2) \in \mathcal{E}[\![\mathit{list}\ \alpha]\!]_\eta) \qquad \textbf{(2)}$$

where η is $\alpha \mapsto (\rho_1, \rho_2, R)$. We may choose R to be $\langle f \rangle$ for some f.

We have (1). Indeed, for all (V_1, V_2) and (W_1, W_2) in $\langle f \rangle$, we have $f\ V_1 \Downarrow V_1$ and $f\ W_1 \Downarrow W_1$, hence $\mathit{cp}_2\ (f\ V_1)(f\ W_1) \Downarrow \mathit{cp}_1\ V_2 W_2$. Thus $\mathit{cp}_2\ (f\ V_1)(f\ W_1) \cong \mathit{cp}_1\ V_2 W_2$. With (H), this implies $\mathit{cp}_2\ V_1 W_1 \cong \mathit{cp}_1\ V_2 W_2$, *i.e.* $\mathit{cp}_2\ V_1 W_1 \sim \mathit{cp}_1\ V_2 W_2$ since we are at type B, as expected.

Applications $\mathit{sort} : \forall \alpha. (\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\ \alpha$

Proof: Assume $\forall x, y,\ \mathit{cp}\ (f\ x)\ (f\ y) \cong \mathit{cp}\ x\ y)$ (**H**).

We have $\mathit{sort} \sim_\sigma \mathit{sort}$ where σ is $\forall \alpha. (\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\ \alpha \to \mathit{list}\ \alpha$.

Thus, for all ρ_1, ρ_2, and relations R in $\mathcal{R}(\rho_1, \rho_2)$,

$$\forall (\mathit{cp}_1, \mathit{cp}_2) \in \mathcal{V}[\![\alpha \to \alpha \to \mathsf{B}]\!]_\eta, \qquad \textbf{(1)}$$
$$\forall (V_1, V_2) \in \mathcal{V}[\![\mathit{list}\ \alpha]\!]_\eta,\ \ (\mathit{sort}\ \rho_1\ \mathit{cp}_1\ \ V_1, \mathit{sort}\ \rho_2\ \mathit{cp}_2\ \ V_2) \in \mathcal{E}[\![\mathit{list}\ \alpha]\!]_\eta) \qquad \textbf{(2)}$$

where η is $\alpha \mapsto (\rho_1, \rho_2, R)$. We may choose R to be $\langle f \rangle$ for some f.

We have (1). Indeed, for all (V_1, V_2) and (W_1, W_2) in $\langle f \rangle$, we have $f\ V_1 \Downarrow V_1$ and $f\ W_1 \Downarrow W_1$, hence $\mathit{cp}_2\ (f\ V_1)(f\ W_1) \Downarrow \mathit{cp}_1\ V_2 W_2$. Thus $\mathit{cp}_2\ (f\ V_1)(f\ W_1) \cong \mathit{cp}_1\ V_2 W_2$. With (H), this implies $\mathit{cp}_2\ V_1 W_1 \cong \mathit{cp}_1\ V_2 W_2$, *i.e.* $\mathit{cp}_2\ V_1 W_1 \sim \mathit{cp}_1\ V_2 W_2$ since we are at type B, as expected. Hence (2) holds.

Since

$$\mathcal{V}[\![\mathit{list}\ \alpha]\!]_\eta \ \triangleq\ \langle \mathit{map}\ \rho_1\ \rho_2\ f \rangle \ \subseteq\ \mathcal{V}[\![\rho_1]\!] \times \mathcal{V}[\![\rho_2]\!]$$

Applications $\mathit{sort} : \forall\alpha.\,(\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\,\alpha$

Proof: Assume $\forall x, y,\ \mathit{cp}\,(f\ x)\,(f\ y) \cong \mathit{cp}\ x\ y)$ (**H**).

We have $\mathit{sort} \sim_\sigma \mathit{sort}$ where σ is $\forall\alpha.\,(\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\,\alpha \to \mathit{list}\,\alpha$.

Thus, for all ρ_1, ρ_2, and relations R in $\mathcal{R}(\rho_1, \rho_2)$,

$$\forall (\mathit{cp}_1, \mathit{cp}_2) \in \mathcal{V}[\![\alpha \to \alpha \to \mathsf{B}]\!]_\eta, \quad \textbf{(1)}$$
$$\forall (V_1, V_2) \in \mathcal{V}[\![\mathit{list}\,\alpha]\!]_\eta,\ \ (\mathit{sort}\ \rho_1\ \mathit{cp}_1\ \ V_1, \mathit{sort}\ \rho_2\ \mathit{cp}_2\ \ V_2) \in \mathcal{E}[\![\mathit{list}\,\alpha]\!]_\eta) \quad \textbf{(2)}$$

where η is $\alpha \mapsto (\rho_1, \rho_2, R)$. We may choose R to be $\langle f \rangle$ for some f.

We have (1). Indeed, for all (V_1, V_2) and (W_1, W_2) in $\langle f \rangle$, we have $f\ V_1 \Downarrow V_1$ and $f\ W_1 \Downarrow W_1$, hence $\mathit{cp}_2\ (f\ V_1)(f\ W_1) \Downarrow \mathit{cp}_1\ V_2 W_2$. Thus $\mathit{cp}_2\ (f\ V_1)(f\ W_1) \cong \mathit{cp}_1\ V_2 W_2$. With (H), this implies $\mathit{cp}_2\ V_1 W_1 \cong \mathit{cp}_1\ V_2 W_2$, *i.e.* $\mathit{cp}_2\ V_1 W_1 \sim \mathit{cp}_1\ V_2 W_2$ since we are at type B, as expected. Hence (2) holds.
Since

$$\mathcal{V}[\![\mathit{list}\,\alpha]\!]_\eta \ \triangleq\ \langle \mathit{map}\ \rho_1\ \rho_2\ f \rangle\ \subseteq\ \mathcal{V}[\![\rho_1]\!] \times \mathcal{V}[\![\rho_2]\!]$$

(2) reads

$$\forall V_1 : \mathit{list}\ \rho_1, V_2 :: \mathit{list}\ \rho_2,\quad \mathit{map}\ \rho_1\ \rho_2\ f\ V_1 \Downarrow V_2 \implies \exists W_1, W_2, \begin{cases} \mathit{map}\ \rho_1\ \rho_2\ f\ W_1 \Downarrow W_2 \\ \mathit{sort}\ \rho_1\ \mathit{cp}_1\ V_1 \Downarrow W_1 \\ \mathit{sort}\ \rho_2\ \mathit{cp}_2\ V_2 \Downarrow W_2 \end{cases}$$

Applications $\quad$ *sort* $: \forall\alpha.\,(\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\,\alpha$

Proof: Assume $\forall x, y,\ \mathit{cp}\,(f\ x)\,(f\ y) \cong \mathit{cp}\ x\ y)$ (**H**).

We have $\mathit{sort} \sim_\sigma \mathit{sort}$ where σ is $\forall\alpha.\,(\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\,\alpha \to \mathit{list}\,\alpha$.

Thus, for all ρ_1, ρ_2, and relations R in $\mathcal{R}(\rho_1, \rho_2)$,

$$\forall (\mathit{cp}_1, \mathit{cp}_2) \in \mathcal{V}[\![\alpha \to \alpha \to \mathsf{B}]\!]_\eta, \qquad \textbf{(1)}$$
$$\forall (V_1, V_2) \in \mathcal{V}[\![\mathit{list}\,\alpha]\!]_\eta,\ (\mathit{sort}\ \rho_1\ \mathit{cp}_1\ \ V_1, \mathit{sort}\ \rho_2\ \mathit{cp}_2\ \ V_2) \in \mathcal{E}[\![\mathit{list}\,\alpha]\!]_\eta) \qquad \textbf{(2)}$$

where η is $\alpha \mapsto (\rho_1, \rho_2, R)$. We may choose R to be $\langle f \rangle$ for some f.

We have (1). Indeed, for all (V_1, V_2) and (W_1, W_2) in $\langle f \rangle$, we have $f\ V_1 \Downarrow V_1$ and $f\ W_1 \Downarrow W_1$, hence $\mathit{cp}_2\ (f\ V_1)(f\ W_1) \Downarrow \mathit{cp}_1\ V_2 W_2$. Thus $\mathit{cp}_2\ (f\ V_1)(f\ W_1) \cong \mathit{cp}_1\ V_2 W_2$. With (H), this implies $\mathit{cp}_2\ V_1 W_1 \cong \mathit{cp}_1\ V_2 W_2$, *i.e.* $\mathit{cp}_2\ V_1 W_1 \sim \mathit{cp}_1\ V_2 W_2$ since we are at type B, as expected. Hence (2) holds.
Since

$$\mathcal{V}[\![\mathit{list}\,\alpha]\!]_\eta \ \triangleq\ \langle \mathit{map}\ \rho_1\ \rho_2\ f \rangle \ \subseteq\ \mathcal{V}[\![\rho_1]\!] \times \mathcal{V}[\![\rho_2]\!]$$

(2) implies

$$\forall V_1 : \mathit{list}\ \rho_1, V_2 :: \mathit{list}\ \rho_2,\quad \mathit{map}\ \rho_1\ \rho_2\ f\ V_1 \Downarrow V_2 \implies \exists W_1, W_2, \begin{cases} \mathit{map}\ \rho_1\ \rho_2\ f\ W_1 \Downarrow W_2 \\ \mathit{sort}\ \rho_1\ \mathit{cp}_1\ \ V_1 \Downarrow W_1 \\ \mathit{sort}\ \rho_2\ \mathit{cp}_2\ \ V_2 \Downarrow W_2 \end{cases}$$

Applications $\mathit{sort} : \forall\alpha.\,(\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\ \alpha$

Proof: Assume $\forall x, y,\ \ \mathit{cp}\ (f\ x)\ (f\ y) \cong \mathit{cp}\ \ x\ y)$ (**H**).

We have $\mathit{sort} \sim_\sigma \mathit{sort}$ where σ is $\forall\alpha.\,(\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\ \alpha \to \mathit{list}\ \alpha$.

Thus, for all ρ_1, ρ_2, and relations R in $\mathcal{R}(\rho_1, \rho_2)$,

$$\forall(\mathit{cp}_1, \mathit{cp}_2) \in \mathcal{V}[\![\alpha \to \alpha \to \mathsf{B}]\!]_\eta, \qquad (\mathbf{1})$$
$$\forall(V_1, V_2) \in \mathcal{V}[\![\mathit{list}\ \alpha]\!]_\eta,\ \ (\mathit{sort}\ \rho_1\ \mathit{cp}_1\ \ V_1, \mathit{sort}\ \rho_2\ \mathit{cp}_2\ \ V_2) \in \mathcal{E}[\![\mathit{list}\ \alpha]\!]_\eta) \qquad (\mathbf{2})$$

where η is $\alpha \mapsto (\rho_1, \rho_2, R)$. We may choose R to be $\langle f \rangle$ for some f.

We have (1). Indeed, for all (V_1, V_2) and (W_1, W_2) in $\langle f \rangle$, we have $f\ V_1 \Downarrow V_1$ and $f\ W_1 \Downarrow W_1$, hence $\mathit{cp}_2\ (f\ V_1)(f\ W_1) \Downarrow \mathit{cp}_1\ V_2 W_2$. Thus $\mathit{cp}_2\ (f\ V_1)(f\ W_1) \cong \mathit{cp}_1\ V_2 W_2$. With (H), this implies $\mathit{cp}_2\ V_1 W_1 \cong \mathit{cp}_1\ V_2 W_2$, *i.e.* $\mathit{cp}_2\ V_1 W_1 \sim \mathit{cp}_1\ V_2 W_2$ since we are at type B, as expected. Hence (2) holds.
Since

$$\mathcal{V}[\![\mathit{list}\ \alpha]\!]_\eta \ \triangleq\ \langle \mathit{map}\ \rho_1\ \rho_2\ f \rangle \ \subseteq\ \mathcal{V}[\![\rho_1]\!] \times \mathcal{V}[\![\rho_2]\!]$$

(2) implies

$$\forall V_1 : \mathit{list}\ \rho_1 \qquad , \qquad \exists W_1, W_2,\ \begin{cases} \mathit{map}\ \rho_1\ \rho_2\ f\ W_1 \Downarrow W_2 \\ \mathit{sort}\ \rho_1\ \mathit{cp}_1\ V_1 \Downarrow W_1 \\ \mathit{sort}\ \rho_2\ \mathit{cp}_2\ (\mathit{map}\ \rho_1\ \rho_2\ f\ V_1) \Downarrow W_2 \end{cases}$$

Applications $\mathit{sort} : \forall\alpha.\,(\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\,\alpha$

Proof: Assume $\forall x, y,\ \mathit{cp}\,(f\ x)\,(f\ y) \cong \mathit{cp}\ x\ y)$ (**H**).

We have $\mathit{sort} \sim_\sigma \mathit{sort}$ where σ is $\forall\alpha.\,(\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\,\alpha \to \mathit{list}\,\alpha$.

Thus, for all ρ_1, ρ_2, and relations R in $\mathcal{R}(\rho_1, \rho_2)$,

$$\forall(\mathit{cp}_1, \mathit{cp}_2) \in \mathcal{V}[\![\alpha \to \alpha \to \mathsf{B}]\!]_\eta, \tag{1}$$

$$\forall(V_1, V_2) \in \mathcal{V}[\![\mathit{list}\,\alpha]\!]_\eta,\ \ (\mathit{sort}\ \rho_1\ \mathit{cp}_1\ \ V_1, \mathit{sort}\ \rho_2\ \mathit{cp}_2\ \ V_2) \in \mathcal{E}[\![\mathit{list}\,\alpha]\!]_\eta) \tag{2}$$

where η is $\alpha \mapsto (\rho_1, \rho_2, R)$. We may choose R to be $\langle f \rangle$ for some f.

We have (1). Indeed, for all (V_1, V_2) and (W_1, W_2) in $\langle f \rangle$, we have $f\ V_1 \Downarrow V_1$ and $f\ W_1 \Downarrow W_1$, hence $\mathit{cp}_2\ (f\ V_1)(f\ W_1) \Downarrow \mathit{cp}_1\ V_2 W_2$. Thus $\mathit{cp}_2\ (f\ V_1)(f\ W_1) \cong \mathit{cp}_1\ V_2 W_2$. With (H), this implies $\mathit{cp}_2\ V_1 W_1 \cong \mathit{cp}_1\ V_2 W_2$, *i.e.* $\mathit{cp}_2\ V_1 W_1 \sim \mathit{cp}_1\ V_2 W_2$ since we are at type B, as expected. Hence (2) holds.

Since

$$\mathcal{V}[\![\mathit{list}\,\alpha]\!]_\eta \ \ \triangleq\ \ \langle \mathit{map}\ \rho_1\ \rho_2\ f \rangle\ \ \subseteq\ \ \mathcal{V}[\![\rho_1]\!] \times \mathcal{V}[\![\rho_2]\!]$$

(2) implies

$$\forall V_1 : \mathit{list}\ \rho_1 \qquad , \qquad \exists W_1, W_2,\ \begin{cases} \mathit{map}\ \rho_1\ \rho_2\ f\ W_1 \Downarrow W_2 \\ \mathit{sort}\ \rho_1\ \mathit{cp}_1\ V_1 \Downarrow W_1 \\ \mathit{sort}\ \rho_2\ \mathit{cp}_2\ (\mathit{map}\ \rho_1\ \rho_2\ f\ V_1) \Downarrow W_2 \end{cases}$$

Applications $\quad$ $\mathit{sort} : \forall\alpha.\,(\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\,\alpha$

Proof: Assume $\forall x, y,\ \mathit{cp}\,(f\ x)\,(f\ y) \cong \mathit{cp}\ x\ y)$ (**H**).

We have $\mathit{sort} \sim_\sigma \mathit{sort}$ where σ is $\forall\alpha.\,(\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\,\alpha \to \mathit{list}\,\alpha$.

Thus, for all ρ_1, ρ_2, and relations R in $\mathcal{R}(\rho_1, \rho_2)$,

$$\forall(\mathit{cp}_1, \mathit{cp}_2) \in \mathcal{V}[\![\alpha \to \alpha \to \mathsf{B}]\!]_\eta, \qquad \textbf{(1)}$$

$$\forall(V_1, V_2) \in \mathcal{V}[\![\mathit{list}\,\alpha]\!]_\eta,\ \ (\mathit{sort}\ \rho_1\ \mathit{cp}_1\ \ V_1, \mathit{sort}\ \rho_2\ \mathit{cp}_2\ \ V_2) \in \mathcal{E}[\![\mathit{list}\,\alpha]\!]_\eta) \qquad \textbf{(2)}$$

where η is $\alpha \mapsto (\rho_1, \rho_2, R)$. We may choose R to be $\langle f\rangle$ for some f.

We have (1). Indeed, for all (V_1, V_2) and (W_1, W_2) in $\langle f\rangle$, we have $f\ V_1 \Downarrow V_1$ and $f\ W_1 \Downarrow W_1$, hence $\mathit{cp}_2\ (f\ V_1)(f\ W_1) \Downarrow \mathit{cp}_1\ V_2 W_2$. Thus $\mathit{cp}_2\ (f\ V_1)(f\ W_1) \cong \mathit{cp}_1\ V_2 W_2$. With (H), this implies $\mathit{cp}_2\ V_1 W_1 \cong \mathit{cp}_1\ V_2 W_2$, *i.e.* $\mathit{cp}_2\ V_1 W_1 \sim \mathit{cp}_1\ V_2 W_2$ since we are at type B, as expected. Hence (2) holds.

Since

$$\mathcal{V}[\![\mathit{list}\,\alpha]\!]_\eta \ \triangleq\ \langle \mathit{map}\ \rho_1\ \rho_2\ f\rangle \ \subseteq\ \mathcal{V}[\![\rho_1]\!] \times \mathcal{V}[\![\rho_2]\!]$$

(2) implies

$$\forall V_1 : \mathit{list}\ \rho_1 \quad , \qquad \exists \quad W_2, \begin{cases} \mathit{map}\ \rho_1\ \rho_2\ f\ (\mathit{sort}\ \rho_1\ \mathit{cp}_1\ V_1) \Downarrow W_2 \\ \mathit{sort}\ \rho_2\ \mathit{cp}_2\ (\mathit{map}\ \rho_1\ \rho_2\ f\ V_1) \Downarrow W_2 \end{cases}$$

Applications $sort : \forall\alpha.\, (\alpha \to \alpha \to bool) \to list\,\alpha$

Proof: Assume $\forall x, y,\ \ cp\,(f\ x)\ (f\ y) \cong cp\ \ x\ y)$ (**H**).

We have $sort \sim_\sigma sort$ where σ is $\forall\alpha.\,(\alpha \to \alpha \to bool) \to list\,\alpha \to list\,\alpha$.

Thus, for all ρ_1, ρ_2, and relations R in $\mathcal{R}(\rho_1, \rho_2)$,

$$\forall (cp_1, cp_2) \in \mathcal{V}[\![\alpha \to \alpha \to \mathsf{B}]\!]_\eta, \qquad \textbf{(1)}$$
$$\forall (V_1, V_2) \in \mathcal{V}[\![list\,\alpha]\!]_\eta,\ \ (sort\,\rho_1\ cp_1\ \ V_1, sort\,\rho_2\ cp_2\ \ V_2) \in \mathcal{E}[\![list\,\alpha]\!]_\eta) \qquad \textbf{(2)}$$

where η is $\alpha \mapsto (\rho_1, \rho_2, R)$. We may choose R to be $\langle f\rangle$ for some f.

We have (1). Indeed, for all (V_1, V_2) and (W_1, W_2) in $\langle f\rangle$, we have $f\ V_1 \Downarrow V_1$ and $f\ W_1 \Downarrow W_1$, hence $cp_2\ (f\ V_1)(f\ W_1) \Downarrow cp_1\ V_2 W_2$. Thus $cp_2\ (f\ V_1)(f\ W_1) \cong cp_1\ V_2 W_2$. With (H), this implies $cp_2\ V_1 W_1 \cong cp_1\ V_2 W_2$, *i.e.* $cp_2\ V_1 W_1 \sim cp_1\ V_2 W_2$ since we are at type B, as expected. Hence (2) holds.

Since

$$\mathcal{V}[\![list\,\alpha]\!]_\eta \ \ \triangleq\ \ \langle map\,\rho_1\,\rho_2\,f\rangle\ \subseteq\ \mathcal{V}[\![\rho_1]\!] \times \mathcal{V}[\![\rho_2]\!]$$

(2) implies

$$\forall V_1 : list\,\rho_1\quad ,\qquad \begin{array}{c} map\,\rho_1\,\rho_2\,f\ (sort\,\rho_1\ cp_1\ V_1) \\ \cong \\ sort\,\rho_2\ cp_2\ (map\,\rho_1\,\rho_2\,f\ V_1) \end{array}$$

Applications $\mathit{sort} : \forall\alpha.\,(\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\,\alpha$

Proof: Assume $\forall x, y,\ \mathit{cp}\,(f\ x)\,(f\ y) \cong \mathit{cp}\ x\ y$) (**H**).

We have $\mathit{sort} \sim_\sigma \mathit{sort}$ where σ is $\forall\alpha.\,(\alpha \to \alpha \to \mathit{bool}) \to \mathit{list}\,\alpha \to \mathit{list}\,\alpha$.

Thus, for all ρ_1, ρ_2, and relations R in $\mathcal{R}(\rho_1, \rho_2)$,

$$\forall (\mathit{cp}_1, \mathit{cp}_2) \in \mathcal{V}[\![\alpha \to \alpha \to \mathsf{B}]\!]_\eta, \qquad \textbf{(1)}$$
$$\forall (V_1, V_2) \in \mathcal{V}[\![\mathit{list}\,\alpha]\!]_\eta,\ (\mathit{sort}\,\rho_1\ \mathit{cp}_1\ \ V_1, \mathit{sort}\,\rho_2\ \mathit{cp}_2\ \ V_2) \in \mathcal{E}[\![\mathit{list}\,\alpha]\!]_\eta) \qquad \textbf{(2)}$$

where η is $\alpha \mapsto (\rho_1, \rho_2, R)$. We may choose R to be $\langle f\rangle$ for some f.

We have (1). Indeed, for all (V_1, V_2) and (W_1, W_2) in $\langle f\rangle$, we have $f\ V_1 \Downarrow V_1$ and $f\ W_1 \Downarrow W_1$, hence $\mathit{cp}_2\ (f\ V_1)(f\ W_1) \Downarrow \mathit{cp}_1\ V_2 W_2$. Thus $\mathit{cp}_2\ (f\ V_1)(f\ W_1) \cong \mathit{cp}_1\ V_2 W_2$. With (H), this implies $\mathit{cp}_2\ V_1 W_1 \cong \mathit{cp}_1\ V_2 W_2$, *i.e.* $\mathit{cp}_2\ V_1 W_1 \sim \mathit{cp}_1\ V_2 W_2$ since we are at type B, as expected. Hence (2) holds.
Since

$$\mathcal{V}[\![\mathit{list}\,\alpha]\!]_\eta \ \triangleq\ \langle \mathit{map}\,\rho_1\,\rho_2\,f\rangle \ \subseteq\ \mathcal{V}[\![\rho_1]\!] \times \mathcal{V}[\![\rho_2]\!]$$

(2) implies

$$\forall V\ : \mathit{list}\,\rho_1 \qquad , \qquad \begin{array}{c} \mathit{map}\ \rho_1\,\rho_2\,f\ (\mathit{sort}\ \rho_1\ \mathit{cp}_1\ V\ ) \\ \cong \\ \mathit{sort}\ \rho_2\ \mathit{cp}_2\ (\mathit{map}\ \rho_1\,\rho_2\ f\ V\ ) \end{array}$$

Applications *whoami* : $\forall\alpha.\, \mathit{list}\ \alpha \to \mathit{list}\ \alpha$

Left as an exercise. . .

Existential types

We define:

$$\mathcal{V}[\![\exists\alpha.\,\tau]\!]_\eta \triangleq \Big\{(\textit{pack}\ V_1, \rho_1\ \textit{as}\ \exists\alpha.\,\tau, \textit{pack}\ V_2, \rho_2\ \textit{as}\ \exists\alpha.\,\tau) \mid \exists\rho_1, \rho_2, R \in \mathcal{R}(\rho_1, \rho_2),\ \ (V_1, V_2) \in \mathcal{E}[\![\tau]\!]_{\eta,\alpha\mapsto(\rho_1,\rho_2,R)}\Big\}$$

Compare with

$$\mathcal{V}[\![\forall\alpha.\,\tau]\!]_\eta = \Big\{(\Lambda\alpha.\,M_1, \Lambda\alpha.\,M_2) \mid \forall\rho_1, \rho_2, R \in \mathcal{R}(\rho_1, \rho_2),\ ((\Lambda\alpha.\,M_1)\ \rho_1, (\Lambda\alpha.\,M_2)\ \rho_2) \in \mathcal{E}[\![\tau]\!]_{\eta,\alpha\mapsto(\rho_1,\rho_2,R)}\Big\}$$

Existential types

Example

Consider $V_1 \triangleq (\mathit{not}, \mathsf{tt})$, and $V_2 \triangleq (\mathit{succ}, 0)$ and $\sigma \triangleq (\alpha \to \alpha) \times \alpha$.
Let $R \in \mathcal{R}(\mathit{bool}, \mathit{nat})$ be $\{(\mathsf{tt}, 2n), (\mathsf{ff}, 2n+1) \mid n \in \mathbb{N}\}$ and η be $\alpha \mapsto (\mathit{bool}, \mathit{nat}, R)$.

We have $(V_1, V_2) \in \mathcal{V}[\![\sigma]\!]_\eta$.

Hence, $(\mathit{pack}\, V_1, \mathit{bool}\ \mathit{as}\ \exists\alpha.\,\sigma,\ \mathit{pack}\, V_2, \mathit{nat}\ \mathit{as}\ \exists\alpha.\,\sigma) \in \mathcal{V}[\![\exists\alpha.\,\sigma]\!]$.

Existential types Example

Consider $V_1 \triangleq (\textit{not}, \mathsf{tt})$, and $V_2 \triangleq (\textit{succ}, 0)$ and $\sigma \triangleq (\alpha \to \alpha) \times \alpha$.
Let $R \in \mathcal{R}(\textit{bool}, \textit{nat})$ be $\{(\mathsf{tt}, 2n), (\mathsf{ff}, 2n+1) \mid n \in \mathbb{N}\}$ and η be $\alpha \mapsto (\textit{bool}, \textit{nat}, R)$.

We have $(V_1, V_2) \in \mathcal{V}[\![\sigma]\!]_\eta$.

Hence, $(\textit{pack}\, V_1, \textit{bool as}\, \exists\alpha.\, \sigma,\ \textit{pack}\, V_2, \textit{nat as}\, \exists\alpha.\, \sigma) \in \mathcal{V}[\![\exists\alpha.\, \sigma]\!]$.

Proof of $\big((\textit{not}, \mathsf{tt}), (\textit{succ}, 0)\big) \in \mathcal{V}[\![(\alpha \to \alpha) \times \alpha]\!]_\eta$ **(1)**

?

Existential types — Example

Consider $V_1 \triangleq (\textit{not}, \mathsf{tt})$, and $V_2 \triangleq (\textit{succ}, 0)$ and $\sigma \triangleq (\alpha \to \alpha) \times \alpha$.
Let $R \in \mathcal{R}(\textit{bool}, \textit{nat})$ be $\{(\mathsf{tt}, 2n), (\mathsf{ff}, 2n+1) \mid n \in \mathbb{N}\}$ and η be $\alpha \mapsto (\textit{bool}, \textit{nat}, R)$.

We have $(V_1, V_2) \in \mathcal{V}[\![\sigma]\!]_\eta$.

Hence, $(\textit{pack}\, V_1, \textit{bool as}\ \exists\alpha.\,\sigma,\ \textit{pack}\, V_2, \textit{nat as}\ \exists\alpha.\,\sigma) \in \mathcal{V}[\![\exists\alpha.\,\sigma]\!]$.

Proof of $\big((\textit{not}, \mathsf{tt}), (\textit{succ}, 0)\big) \in \mathcal{V}[\![(\alpha \to \alpha) \times \alpha]\!]_\eta$ **(1)**

We have $(\mathsf{tt}, 0) \in \mathcal{V}[\![\alpha]\!]_\eta$, since $(\mathsf{tt}, 0) \in R$.
We also have $(\textit{not}, \textit{succ}) \in \mathcal{V}[\![\alpha \to \alpha]\!]_\eta$, which proves (1).

?

Existential types Example

Consider $V_1 \triangleq (\textit{not}, \mathsf{tt})$, and $V_2 \triangleq (\textit{succ}, 0)$ and $\sigma \triangleq (\alpha \to \alpha) \times \alpha$.
Let $R \in \mathcal{R}(\textit{bool}, \textit{nat})$ be $\{(\mathsf{tt}, 2n), (\mathsf{ff}, 2n+1) \mid n \in \mathbb{N}\}$ and η be $\alpha \mapsto (\textit{bool}, \textit{nat}, R)$.

We have $(V_1, V_2) \in \mathcal{V}[\![\sigma]\!]_\eta$.

Hence, $(\textit{pack}\, V_1, \textit{bool as}\, \exists\alpha.\,\sigma,\ \textit{pack}\, V_2, \textit{nat as}\, \exists\alpha.\,\sigma) \in \mathcal{V}[\![\exists\alpha.\,\sigma]\!]$.

Proof of $\big((\textit{not}, \mathsf{tt}), (\textit{succ}, 0)\big) \in \mathcal{V}[\![(\alpha \to \alpha) \times \alpha]\!]_\eta$ $\mathbf{(1)}$

We have $(\mathsf{tt}, 0) \in \mathcal{V}[\![\alpha]\!]_\eta$, since $(\mathsf{tt}, 0) \in R$.
We also have $(\textit{not}, \textit{succ}) \in \mathcal{V}[\![\alpha \to \alpha]\!]_\eta$, which proves (1).

Indeed, assume $(W_1, W_2) \in \mathcal{V}[\![\alpha]\!]_\eta$. Then (W_1, W_2) is either of the form

- $(\mathsf{tt}, 2n)$ and $(\textit{not}\, W_1, \textit{succ}\, W_2)$ reduces to $(\mathsf{ff}, 2n+1)$, or
- $(\mathsf{ff}, 2n+1)$ and $(\textit{not}\, W_1, \textit{succ}\, W_2)$ reduces to $(\mathsf{tt}, 2n+2)$.

In both cases, $(\textit{not}\, W_1, \textit{succ}\, W_2)$ reduces to a pair in R.
Hence, $(\textit{not}\, W_1, \textit{succ}\, W_2) \in \mathcal{E}[\![\alpha]\!]_\eta$.

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 ρ_1 and ρ_2.

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

Assume that ρ_1 and ρ_2 are two closed representation types and R is in $\mathcal{R}(\rho_1, \rho_2)$. Let η be $\alpha \mapsto (\rho_1, \rho_2, R)$.

Suppose that $N_1 : \tau[\alpha \mapsto \rho_1]$ and $N_2 : \tau[\alpha \mapsto \rho_2]$ are two equivalent implementations of the operations, *i.e.* such that $(N_1, N_2) \in \mathcal{E}[\![\tau]\!]_\eta$.

A client M satisfies $(M, M) \in \mathcal{E}[\![\forall\alpha.\tau \to \sigma]\!]_\eta$. Thus $(M\ \rho_1\ N_1, M\ \rho_2\ N_2)$ is in $\mathcal{E}[\![\sigma]\!]$ (as α is not free in σ).

That is, $M\ \rho_1\ N_1 \cong_\sigma M\ \rho_2\ N_2$: the behavior with the implementation N_1 with representation type ρ_1 is indistinguishable from the behavior with the implementation N_2 with representation type ρ_2.

How do we deal with recursive types?

Assume that we allow equi-recursive types.

$$\tau ::= \ldots \mid \mu\alpha.\tau$$

A naive definition would be

$$\mathcal{V}[\![\mu\alpha.\tau]\!]_\eta = \mathcal{V}[\![[\alpha \mapsto \mu\alpha.\tau]\tau]\!]_\eta$$

But this is ill-founded.

The solution is to use indexed-logical relations.

We use a sequence of decreasing relations indexed by integers (fuel), which is consumed during unfolding of recursive types.

Step-indexed logical relations (a taste)

We define a sequence $\mathcal{V}_k[\![\tau]\!]_\eta$ indexed by natural numbers $n \in \mathbb{N}$ that relates values of type τ up to n reduction steps. Omitting typing clauses:

$$\begin{aligned}
\mathcal{V}_k[\![\mathsf{B}]\!]_\eta &= \{(\mathsf{tt},\mathsf{tt}),(\mathsf{ff},\mathsf{ff})\} \\
\mathcal{V}_k[\![\tau \to \sigma]\!]_\eta &= \{(V_1,V_2) \mid \forall j < k, \forall (W_1,W_2) \in \mathcal{V}_j[\![\tau]\!]_\eta, \\
&\qquad\qquad (V_1\ W_1, V_2\ W_2) \in \mathcal{E}_j[\![\sigma]\!]_\eta\} \\
\mathcal{V}_k[\![\alpha]\!]_\eta &= \eta_R(\alpha).k \\
\mathcal{V}_k[\![\forall\alpha.\,\tau]\!]_\eta &= \big\{(V_1,V_2) \mid \forall \rho_1,\rho_2, R \in \mathcal{R}^k(\rho_1,\rho_2), \forall j < k, \\
&\qquad\qquad (V_1\ \rho_1, V_2\ \rho_2) \in \mathcal{V}_j[\![\tau]\!]_{\eta,\alpha\mapsto(\rho_1,\rho_2,R)}\big\} \\
\mathcal{V}_k[\![\mu\alpha.\tau]\!]_\eta &= \mathcal{V}_{k-1}[\![[\alpha \mapsto \mu\alpha.\tau]\tau]\!]_\eta \\
\mathcal{E}_k[\![\tau]\!]_\eta &= \{(M_1,M_2) \mid \forall j < k, M_1 \Downarrow_j V_1 \\
&\qquad \Longrightarrow \exists V_2, M_2 \Downarrow V_2 \wedge (V_1,V_2) \in \mathcal{V}_{k-j}[\![\tau]\!]_\eta\}
\end{aligned}$$

By $\Downarrow_j$ means *reduces in j-steps*.
$\mathcal{R}^j(\rho_1,\rho_2)$ is composed of sequences of decreasing relations between closed values of closed types ρ_1 and ρ_2 of length (at least) j.

Step-indexed logical relations (a taste)

The relation is asymmetric.

If $\Delta;\Gamma \vdash M_1, M_2 : \tau$ we define $\Delta;\Gamma \vdash M_1 \precsim M_2 : \tau$ as

$$\forall \eta \in \mathcal{R}^k_\Delta(\delta_1, \delta_2), \forall (\gamma_1, \gamma_2) \in \mathcal{G}_k[\![\Gamma]\!],\ \ (\gamma_1(\delta_1(M_1)), \gamma_2(\delta_2(M_2)) \in \mathcal{E}_k[\![\tau]\!]_\eta$$

and

$$\Delta;\Gamma \vdash M_1 \sim M_2 : \tau \ \ \triangleq\ \ \bigwedge \begin{cases} \Delta;\Gamma \vdash M_1 \precsim M_2 : \tau \\ \Delta;\Gamma \vdash M_2 \precsim M_1 : \tau \end{cases}$$

Notations and proofs get a bit involved...

Notations may be simplified by introducing a *later* guard $\triangleright$ to capture incrementation of the index and avoid the explicit manipulation of integers (but the meaning remains the same).

Logical relations for F^ω ?

Logical relations can be generalized to work for F^ω, indeed.

There is a slight complication though in the interpretation of type functions.

This is out of this course scope, but one may, for instance, read [Atkey, 2012].

Bibliography I

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

▷ Robert Atkey. Relational parametricity for higher kinds. In Patrick Cégielski and Arnaud Durand, editors, *Computer Science Logic (CSL'12) - 26th International Workshop/21st Annual Conference of the EACSL*, volume 16 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 46–61, 2012. doi: 10.4230/LIPIcs.CSL.2012.46.

▷ Jean-Philippe Bernardy, Patrik Jansson, and Koen Claessen. *Testing Polymorphic Properties*, pages 125–144. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-11957-6. doi: $10.1007/978\text{-}\sigma_2 3\text{-}\sigma_2 642\text{-}\sigma_2 11957\text{-}\sigma_2 6\_8$.

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.

Bibliography II

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.

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

▷ Philip Wadler. The Girard-Reynolds isomorphism (second edition). *Theoretical Computer Science*, 375(1–3):201–226, May 2007.