MPRI 2.4, Functional programming and type systems
Metatheory of System F

Didier Rémy

November 03, 2020

INRIA
Plan of the course

Metatheory of System F

ADTs, Existential types, GATDs

Logical relations
Messages

Last lesson Friday, November 22

Partial exams on Friday, November 29
  • Only course notes and handwritten notes are allowed.
  • No electronic devices of any kind.
  • Bring you own paper (preferably *double* A4 sheet)

Internships — see the course webpage
  • Sharing and Unsharing in Hindley Milner
  • Propagation of type annotations in Hindley-Milner based type-systems
  • *Verifying Chunked Sequences, by F. Pottier (and A. Charguéraud)*
  • *Plus 2 others by Yann Régis Giannas.*
  (Talk to me if you are interested)
Logical relations and parametricity
Contents

- Introduction
- Normalization of $\lambda_{st}$
- Observational equivalence in $\lambda_{st}$
- Logical relations in stlc
- Logical relations in F
- Applications
- Extensions
What are logical relations?

So far, most proofs involving terms have proceeded by induction on the structure of *terms* (or, equivalently, *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
- They can be used to prove type safety and strong normalization

**Binary relations**

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

Logical relations are a common proof method for programming languages.
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 $\forall \alpha. \alpha \rightarrow \text{int}$?

- the function cannot examine its argument
- it always return 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 a term of the form $\lambda x. n$
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 $\forall \alpha. \alpha \rightarrow \text{int}$?

$\triangleright$ behaves as $\lambda x. n$

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

$\triangleright$ behaves as $\lambda x. x$

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

$\triangleright$ behaves as $\lambda x. \lambda y. x$

A term type $\forall \alpha. \alpha \rightarrow \alpha \rightarrow \alpha$?

$\triangleright$ behaves either as $\lambda x. \lambda y. x$ or $\lambda x. \lambda y. y$
Pametricity

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

What properties may we learn from a function

$$\text{whoami} : \forall \alpha. \text{list } \alpha \to \text{list } \alpha$$

- 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

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

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

What properties may we learn from a function

\[ \text{whoami} : \forall \alpha. \text{list } \alpha \to \text{list } \alpha \]

What property may we learn for the list sorting function?

\[ \text{sort} : \forall \alpha. (\alpha \to \alpha \to \text{bool}) \to \text{list } \alpha \to \text{list } \alpha \]

If \( f \) is order-preserving, then sorting commutes with \( \text{map } f \)

\[
(\forall x, y, \ cmp \ (f \ x) \ (f \ y) = cmp \ x \ y) \implies \\
\forall \ell, \ \text{sort } cmp \ (\text{map } f \ \ell) = \text{map } f \ (\text{sort } cmp \ \ell)
\]
Pametricity

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

What properties may we learn from a function

\[ \text{whoami} : \forall \alpha. \text{list } \alpha \to \text{list } \alpha \]

What property may we learn for the list sorting function?

\[ \text{sort} : \forall \alpha. (\alpha \to \alpha \to \text{bool}) \to \text{list } \alpha \to \text{list } \alpha \]

If \( f \) is order-preserving, then sorting commutes with \( \text{map } f \)

\[ (\forall x, y, \text{cmp}_2 (f x) (f y) = \text{cmp}_1 x y) \implies \forall \ell, \text{sort } \text{cmp}_2 (\text{map } f \ell) = \text{map } f (\text{sort } \text{cmp}_1 \ell) \]

Application:

- If \( \text{sort} \) is correct on lists of integers, then it is correct on any list
- May be useful to reduce testing.
Pametricity

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

What properties may we learn from a function

\[ \text{whoami} : \forall \alpha. \text{list } \alpha \rightarrow \text{list } \alpha \]

What property may we learn for the list sorting function?

\[ \text{sort} : \forall \alpha. (\alpha \rightarrow \alpha \rightarrow \text{bool}) \rightarrow \text{list } \alpha \rightarrow \text{list } \alpha \]

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

\[ (\forall x, y, \ cmp_2 (f \ x) (f \ y) = \ cmp_1 \ x \ y) \implies \forall \ell, \ \text{sort} \ cmp_2 (\text{map} \ f \ \ell) = \text{map} \ f \ (\text{sort} \ cmp_1 \ \ell) \]

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

- Introduction
- Normalization of $\lambda_{st}$
- Observational equivalence in $\lambda_{st}$
- Logical relations in stlc
- Logical relations in F
- Applications
- Extensions
Normalization of simply-typed $\lambda$-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 $\lambda$-calculus is also lifted at the level of types in richer type systems such as System $F^\omega$; 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 $\lambda$-calculus is a simple and illustrative use of logical relations.

*Notice however, that our simply-typed $\lambda$-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 $\lambda$-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].
Tait’s method

Idea

- build the set $\mathcal{T}_\tau$ of terminating terms of type $\tau$;
- show that any term of type $\tau$ is in $\mathcal{T}_\tau$, by induction on terms.

This hypothesis is however too weak. The difficulty is as usual to find a strong enough induction hypothesis...

Terms of type $\tau_1 \to \tau_2$ should not only terminate but also terminate when applied to terms in $\mathcal{T}_{\tau_1}$.

The construction of $\mathcal{T}_\tau$ is thus by induction of $\tau$. 
Normalization

Definition
Let $\mathcal{T}_\tau$ be defined inductively on $\tau$ as follows:

- $\mathcal{T}_\alpha$ is the set of closed terms that terminates;
- $\mathcal{T}_{\tau_2 \rightarrow \tau_1}$ is the set of closed terms $M_1$ of type $\tau_2 \rightarrow \tau_1$ that terminates and such that $M_1 M_2$ is in $\mathcal{T}_{\tau_1}$ for any term $M_2$ in $\mathcal{T}_{\tau_2}$.

The set $\mathcal{T}_\tau$ can be seen as a predicate, i.e. a unary relation. It is called a logical relation because it is defined inductively on the structure of types.

The following proofs is then schematic of the use of logical relations.
Normalization

Reduction of terms of type $\tau$ preserves membership in $T_\tau$ (this is stronger than stability of $T_\tau$ by reduction):

**Lemma**

If $\emptyset \vdash M : \tau$ and $M \rightarrow M'$, then $M \in T_\tau$ iff $M' \in T_\tau$.

**Proof.**

The proof is by induction on $\tau$.

**Lemma**

For any type $\tau$, the reduction of any term in $T_\tau$ terminates.

Tautology, by definition of $T_\tau$. 
Normalization

Therefore, it just remains to show that any term of type $\tau$ is in $\mathcal{T}_\tau$, i.e.:

**Lemma**

If $\emptyset \vdash M : \tau$, then $M \in \mathcal{T}_\tau$.

The proof is by induction on (the typing derivation of) $M$.

However, the case for abstraction requires some similar statement, but for open terms. We need to strengthen the Lemma.

A trick to avoid considering open terms is to require the statement to hold for all closed instances of an open term:

**Lemma (strengthened)**

If $(x_i : \tau_i)_{i \in I} \vdash M : \tau$, then for any closed values $(V_i)_{i \in I}$ in $(\mathcal{T}_{\tau_i})_{i \in I}$, the term $[(x_i \mapsto V_i)_{i \in I}]M$ is in $\mathcal{T}_\tau$. 
Normalization

**Proof.** By structural induction on $M$.
We write $\Gamma$ for $(x_i : \tau_i)_{i \in I}$ and $\theta$ for $[(x_i \mapsto V_i)_{i \in I}]$. Assume $\Gamma \vdash M : \tau$.

The only interesting case is when $M$ is $\lambda x : \tau_1 . M_2$:

By inversion of typing, we know that $\Gamma, x : \tau_1 \vdash M_2 : \tau_2$ and $\tau_1 \rightarrow \tau_2$ is $\tau$.

To show $\theta M \in T_\tau$, we must show that it is terminating, which holds as it is a value, and that its application to any $M_1$ in $T_\tau_1$ is in $T_\tau_2$ (1).

Let $M_1 \in T_\tau_1$. By definition $M_1 \xrightarrow{*} V$ (2). We then have:

\[
(\theta M) M_1 \triangleq (\theta(\lambda x : \tau_1 . M_2)) M_1
= (\lambda x : \tau_1 . \theta M_2) M_1
\xrightarrow{*} (\lambda x : \tau_1 . \theta M_2) V
\xrightarrow{} [x \mapsto V](\theta M_2)
\xrightarrow{} ([x \mapsto V] \theta)(M_2) \in T_\tau_2
\]

This establishes (1) since membership in $T_\tau_2$ is preserved by reduction.
Calculus

Take the call-by-value $\lambda_{st}$ with primitive booleans and conditional. Write $B$ the type of booleans and $tt$ and $ff$ for $true$ and $false$.

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

$V[B] \triangleq \{tt, ff\}$

$V[\tau_1 \rightarrow \tau_2] \triangleq \{\lambda x: \tau_1. M | \lambda x: \tau_1. M : \tau_1 \rightarrow \tau_2$

$\quad \land \forall V \in V[\tau_1], (\lambda x: \tau_1. M) V \in E[\tau_2]\}$

$E[\tau] \triangleq \{M | M : \tau \land \exists V \in V[\tau], M \downarrow V\}$

We write $M \downarrow V$ for $M \rightarrow^* V$.

The goal is to show that any closed expression of type $\tau$ is in $E[\tau]$.

Remarks

$V[\tau] \subseteq E[\tau]$—by definition.

$E[\tau]$ is closed by inverse reduction—by definition, i.e.

If $M : \tau$ and $M \rightarrow N$ and $N \in E[\tau]$ then $M \in E[\tau]$. 

Problem

We wish to show that every closed term of type $\tau$ is in $E[\tau]$

- 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 $\tau$ as *closed values* in $\mathcal{V}[\tau]$.

We interpret environments $\Gamma$ as *closing substitutions* $\gamma$, i.e. mappings from term variables to *closed values*:

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

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

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

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

That is, closed well-typed terms of type $\tau$ evaluates to values of type $\tau$. 
Proof by induction on the typing derivation

Routine cases

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

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

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

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

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

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

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

By (2), there exists $V \in \mathcal{V}[\tau_2]$ such that $\gamma M_2 \Downarrow V$.

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

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

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


Proof by induction on the typing derivation

The interesting case

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

Assume $\gamma \in \mathcal{G}[\Gamma]$.

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

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

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

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

where $\gamma'$ 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. Therefore by (3), we have $\gamma' M \in \mathcal{E}[\tau]$. Since $\mathcal{E}[\tau]$ is closed by inverse reduction, this proves (2) which finishes the proof of (1).
Variations

We have shown both *termination* and *type soundness*, simultaneously. Termination would not hold if we had a fix point. But type soundness would still hold.

The proof may be modified by choosing:

\[
\mathcal{E}[^\tau] = \{ M : \tau \mid \forall N, M \Downarrow N \implies N \in \mathcal{V}[^\tau] \lor \exists N', N \rightarrow N' \}
\]

**Exercise**

*Show type soundness with this semantics.*
Contents

- Introduction
- Normalization of $\lambda_{st}$
- Observational equivalence in $\lambda_{st}$
- Logical relations in stlc
- Logical relations in F
- Applications
- Extensions
(Bibliography)

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

See also

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

We assume a call-by-name semantics for generality of the presentation, but all results also apply to a call-by-value semantics.
When are two programs equivalent

\[ M \Downarrow N \, ? \]

\[ M \Downarrow V \, \text{and} \, N \Downarrow V ? \]

But what if \( M \) and \( N \) are functions?

Aren’t \( \lambda x. (x + x) \) and \( \lambda x. 2 \times x \) equivalent?

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

Are we general enough?
Observational equivalence

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

If $M : B$ and $N : B$, then $M \simeq N$ iff there exists $V$ such that $M \Downarrow V$ 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)**

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

**Typing of contexts**

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

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

We write $M \simeq_{\tau} N$ for $\emptyset \vdash M \simeq N : \tau$
Observational equivalence

Observational equivalence is the coarsest consistent congruence, where:

- ≡ is consistent if \( \emptyset \vdash M \equiv N : B \) implies \( M \simeq N \).
- ≡ is a congruence if it is an equivalence and is closed by context, i.e.

\[
\Gamma \vdash M \equiv N : \tau \land C : (\Gamma \triangleright \tau) \leadsto (\Delta \triangleright \sigma) \quad \implies \quad \Delta \vdash C[M] \equiv C[N] : \sigma
\]

**Consistent**: by definition, using the empty context.

**Congruence**: by compositionality of contexts.

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

Let \( C : (\Gamma \triangleright \tau) \leadsto (\emptyset \triangleright B) \) (2). We must show that \( C[M] \simeq C[N] \).

This follows by consistency applied to \( \Gamma \vdash C[M] \equiv C[N] : B \) which 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

- Introduction
- Normalization of $\lambda_{st}$
- Observational equivalence in $\lambda_{st}$
- Logical relations in stlc
- Logical relations in F
- Applications
- Extensions
Logical equivalence for closed terms

We inductively define $M \sim_\tau M'$ on closed terms of (ground) type $\tau$ by induction on $\tau$:

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

**Lemma**

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

**Note**

Reflexivity is not obvious at all.
Logical equivalence for closed terms

Proof by induction on type $\tau$

Case $\tau$ is $B$ for values: the result is immediate.

Case $\tau$ is $\tau_1 \rightarrow \tau_2$: By IH, symmetry and transitivity hold at types $\tau_1$ and $\tau_2$.

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

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

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

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

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

(C) follows by transitivity of (C1) and (C2) at type $\tau_2$. 
Properties of logical equivalence

**Closure by inverse reduction**

Assume that $N : \tau$ and $M \sim_\tau M'$.
If $N \Downarrow M$ and $N' \Downarrow M'$ then $N \sim_\tau N'$.

The proof is by induction on $\tau$.
(We show it for a single reduction step, e.g. on the left-hand side)

**Case $\tau$ is $B$:** By closure of behavioral equivalence $\simeq$ by inverse reduction.

**Case $\tau$ is $\tau_1 \to \tau_2$:** To show $N \sim_\tau M'$ we assume $M_1 \sim_{\tau_1} M'_1$ and show $N M_1 \sim_{\tau_2} M' M'_1$ (1).
From $M \sim_\tau M'$, we have $M M_1 \sim_{\tau_2} M' M'_1$. The conclusion (1) then follows by IH at type $\tau_2$, since we have $N M_1 \rightarrow M M_1$ as a consequence of the assumption $N \rightarrow M$.

**Consistency** If $M \sim_B M'$, then $M \simeq M'$
(Obvious, by definition.)
Logical equivalence for open terms

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

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

**Definition**

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

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

**Immediate properties**

Open logical equivalence is symmetric and transitive.

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

Theorem (Reflexivity)

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

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

Case $M$ is $\lambda x: \tau_1. N$ and $\tau$ is $\tau_1 \to \tau_2$ with $x \not\in \gamma, \gamma'$.
We show $\lambda x: \tau_1. \gamma N \sim_{\tau_1 \to \tau_2} \lambda x: \tau_1. \gamma' N$. Assume $M_1 \sim_{\tau_1} M'_1$ (3).
We must show $(\lambda x: \tau_1. \gamma N) M_1 \sim_{\tau_2} (\lambda x: \tau_1. \gamma' N) M'_1$.

By inverse reduction, it suffices to show
$\gamma(N)[x \mapsto M_1] \sim_{\tau_2} \gamma'(N)[x \mapsto M'_1]$, i.e.
$\gamma_1(N) \sim_{\tau_2} \gamma'_1(N)$ where $\gamma_1$ is $(\gamma[x \mapsto M_1])$ and $\gamma'_1$ is $(\gamma'[x \mapsto M'_1])$ (4).

We have $\gamma_1 \sim_{\Gamma, x: \tau_1} \gamma'_1$ (5) from (2) and (3).
By inversion of typing applied to (1), we have $\Gamma, x : \tau_1 \vdash N : \tau_2$.
Thus (4) follows by induction hypothesis applied with (5).
Properties of logical equivalence

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

*Case \( M \) is \( \text{tt} \) or \( \text{ff} \) and \( \tau \) is \( B \):* Since \( M \) is closed it suffices to show \( M \sim_{B} M \) which holds by reflexivity of \( \sim_{B} \), i.e. of behavioral equivalence \( \simeq \).

*Case \( M \) is \( M_1 M_2 \):* By induction hypothesis and the fact that substitution distributes over term application.

*Case \( M \) is \( x \):* Immediate.
Properties of logical equivalence

Proof (continued)

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

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

Otherwise, $\gamma N \Downarrow \text{ff}$, and we proceed symmetrically.
Properties of logical relations

**Corollary (equivalence)** Open logical relation is an equivalence relation.

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

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

**Logical equivalence is a congruence**
If $\Gamma \vdash M \sim M' : \tau$ and $C : (\Gamma \triangleright \tau) \sim (\Delta \triangleright \sigma)$, then $\Delta \vdash C[M] \sim C[M'] : \sigma$.

**Proof** By induction on the proof of $C : (\Gamma \triangleright \tau) \sim (\Delta \triangleright \sigma)$. Similar to the proof of reflexivity. *(We need a definition of context typing derivations by a set of typing rules to be able to reason by induction on the typing derivation.)*

**Corollary** Logical equivalence implies observational equivalence.
If $\Gamma \vdash M \sim M' : \tau$ then $\Gamma \vdash M \equiv M' : \tau$.

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

Lemma
Observational equivalence of closed terms implies logical equivalence.
If $M \equiv_{\tau} M'$ then $M' \sim_{\tau} M'$.

Proof by induction on $\tau$.

Case $\tau$ is $B$: In the empty context, we have $M \approx_{B} M'$, hence $M \sim_{B} M'$.

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

**Corollary (Value arguments)**

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

**Proof**

Assume \( N \sim_{\tau_1} N' \).

There exists \( V \) and \( V' \) such that \( N \Downarrow V \) and \( N' \Downarrow V' \).

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

We have \( N \sim_{\tau_1} V \) from \( N \Downarrow V \) and **closure by inverse reduction**.

Then \( M N \sim_{\tau_2} M V \) follows by **congruence** of \( \sim_{\tau_2} \).

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

The conclusion (1) follows by transitivity of \( \sim_{\tau_2} \) with (H).
Logical equivalence: application

Assume \( \text{not} \triangleq \lambda x:B. \text{if } x \text{ then } \text{ff} \text{ else } \text{tt} \)
and \( M \triangleq \lambda x:B. \lambda y:\tau. \lambda z:\tau. \text{if } \text{not } x \text{ then } y \text{ else } z \)
and \( M' \triangleq \lambda x:B. \lambda y:\tau. \lambda z:\tau. \text{if } x \text{ then } z \text{ else } y \)

Show that \( M \cong_B \tau \rightarrow \tau \rightarrow \tau \) \( M' \) (C).

It suffices to show \( M \ V_0 \ V_1 \ V_2 \sim_\tau M' \ V_0' \ V_1' \ V_2' \) whenever \( V_0 \cong_B V_0' \) and \( V_1 \sim_\tau V_1' \) and \( V_2 \sim_\tau V_2' \).

By inverse reduction, it suffices to show

\[
\text{if not } V_0 \text{ then } V_1 \text{ else } V_2 \sim_\tau \text{ if } V_0' \text{ then } V_2' \text{ else } V_1'
\]

By cases on \( V_0 \).

\textbf{Case } V_0 \text{ is tt}: Then \( \text{not } V_0 \Downarrow \text{ff} \) and thus \( M \Downarrow V_2 \) while \( M' \Downarrow V_2 \). Then (C) follows by inverse reduction and \( V_2 \sim_\tau V_2' \).

\textbf{Case } V_0 \text{ is ff}: is symmetric.
Contents

- Introduction
- Normalization of $\lambda_{st}$
- Observational equivalence in $\lambda_{st}$
- Logical relations in stlc
- Logical relations in F
- Applications
- Extensions
Observational equivalence

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

\[ \tau ::= \ldots | \alpha | \forall \alpha. \tau \quad M ::= \ldots | \Lambda \alpha. M | M \tau \]

We write typing contexts \( \Delta; \Gamma \) where \( \Delta \) binds variables and \( \Gamma \) binds program variables.

Typing of contexts becomes \( C : (\Delta; \Gamma \triangleright \tau) \leadsto (\Delta'; \Gamma' \triangleright \tau') \).

**Observational equivalence**

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

\[ \forall C : (\Delta; \Gamma \triangleright \tau) \leadsto (\emptyset; \emptyset \triangleright B), \; C[M] \cong C[M'] \]

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

For closed terms (no free program variables)

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

For simple types, we defined logical relations and observed that

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

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

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

- It respects observational equivalence: If $R(M, M')$ and $N \equiv_\rho M$ and $N' \equiv_\rho M'$, then $R(N, N')$.
- It is closed under inverse reduction: If $R(M, M')$ and $N \downarrow M$ and $N' \downarrow M'$, then $R(N, N')$.

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

Take

\[ \Delta \overset{\Delta}{=} \alpha \quad \delta \overset{\Delta}{=} \alpha \mapsto B \quad \delta' \overset{\Delta}{=} \alpha \mapsto \mathbb{Z} \]

Then \( R: \delta \leftrightarrow_\alpha \delta' \) may be the closure by inverse reduction (written \( \Diamond \))

\[ \Diamond\{(tt, 0)\} \cup \{(ff, n) \mid n \in \mathbb{Z}^*\} \]

where integers may be used to simulate booleans.

*Allows to relate values at different types.*
Logical equivalence for closed terms with open types

Assume $\eta : \delta \leftrightarrow \Delta \delta'$ and $M : \delta(\tau)$ and $M' : \delta'(\tau)$.

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

\begin{align*}
M \sim_B M' \left[ \eta : \delta \leftrightarrow \delta' \right] & \iff M \simeq M' \\
M \sim_{\tau_1 \to \tau_2} M' \left[ \eta : \delta \leftrightarrow \delta' \right] & \iff \text{for all } N \sim_{\tau_1} N' \left[ \eta : \delta \leftrightarrow \delta' \right], \quad M N \sim_{\tau_2} M' N' \left[ \eta : \delta \leftrightarrow \delta' \right] \\
M \sim_{\alpha} M' \left[ \eta : \delta \leftrightarrow \delta' \right] & \iff \eta(\alpha)(M, M') \\
M \sim_{\forall \alpha \cdot \tau} M' \left[ \eta : \delta \leftrightarrow \delta' \right] & \iff \text{for all } \rho, \rho', R : \rho \leftrightarrow \rho', \quad M \rho \sim_{\tau} M' \rho' \\
& \quad \left[ (\eta, \alpha \mapsto R) : (\delta, \alpha \mapsto \rho) \leftrightarrow (\delta', \alpha \mapsto \rho') \right]
\end{align*}
Logical equivalence for open terms

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

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

where $\gamma \sim_\Gamma \gamma' [\eta : \delta \leftrightarrow \delta'] \triangleq \bigwedge \begin{cases} \text{dom}(\gamma) = \text{dom}(\gamma') = \text{dom}(\Gamma) \\ \forall x : \tau \in \text{dom}(\Gamma), \; \gamma(x) \sim_\tau \gamma'(x) \; [\eta : \delta \leftrightarrow \delta'] \end{cases}$

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

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

Closure under inverse reduction

If \( M \sim_\tau M' \ [\eta : \delta \leftrightarrow \delta'] \) and \( N \downarrow M \) and \( N' \downarrow M' \) (and \( N : \delta(\tau) \) and \( N' : \delta'(\tau) \)), then \( N \sim_\tau N' \ [\eta : \delta \leftrightarrow \delta'] \).

Proof by induction on \( \tau \).

Similar to the monomorphic case, except for:

Case \( \tau \) is \( \forall \alpha. \sigma \):

To show \( N \sim_\tau M' \ [\eta : \delta \leftrightarrow \delta'] \),
i.e. by definition, \( \forall \rho, \rho', R : \rho \leftrightarrow \rho' \), \( M \rho \sim_\tau M' \rho' \ [(\eta, \alpha \mapsto R)] \),
we assume \( R : \rho \leftrightarrow \rho' \) and show \( N \rho \sim_\sigma M' \rho' \ [\eta, \alpha \mapsto R] \).
Since \( N \rho \rightarrow M \rho \),
by induction hypothesis it suffices to show \( M \rho \sim_\sigma M' \rho' \ [\eta, \alpha \mapsto R] \),
which follows from \( M \sim_\tau M' \ [\eta : \delta \leftrightarrow \delta'] \).
Properties

Respect for observational equivalence

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

Proof by induction on $\tau$.

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

Case $\tau$ is $\forall \alpha. \sigma$:

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

Since $N \rho \cong_{\delta(\tau)} M \rho$ (by (2) as $\cong$ is a congruence), by induction hypothesis it suffices to show $M \rho \sim_{\sigma} M' \rho' [\eta, \alpha \mapsto R]$, which follows from (1).
Properties

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

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

Lemma (respect for observational equivalence)
If $\Delta;\Gamma \vdash M \sim M' : \tau$ and $\Delta;\Gamma \vdash M \cong N : \tau$ and $\Delta;\Gamma \vdash M' \cong N' : \tau$, then $\Delta;\Gamma \vdash N \sim N' : \tau$

Lemma (Compositionality)

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

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

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

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

Proof by structural induction on $\tau$. 
Parametricity

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

**Proof** by induction on the typing derivation.
Proof of parametricity

Case $M$ is $\Lambda\alpha.N$: We must show that $\Delta;\Gamma \vdash \Lambda\alpha.N \sim \Lambda\alpha.N : \forall\alpha.\tau$.
Assume $\eta : \delta \leftrightarrow \Delta \delta'$ and $\gamma \approx_{\Gamma} \gamma' [\eta : \delta \leftrightarrow \delta'].$

We must show $\gamma(\delta(\Lambda\alpha.N)) \sim_{\forall\alpha.\tau} \gamma'(\delta(\Lambda\alpha.N)) [\eta : \delta \leftrightarrow \delta'].$

Assume $\sigma$ and $\sigma'$ closed and $R : \sigma \leftrightarrow \sigma'$. We must show
\[
(\gamma(\delta(\Lambda\alpha.N))) \sigma \sim_{\tau} (\gamma'(\delta'(\Lambda\alpha.N))) \sigma [\eta_0 : \delta_0 \leftrightarrow \delta_0']
\]
where $\eta_0 = \eta, \alpha \mapsto R$ and $\delta_0 = \delta, \alpha \mapsto \sigma$ and $\delta_0' = \delta, \alpha \mapsto \sigma'$.
Since
\[
(\gamma(\delta(\Lambda\alpha.N))) \sigma = (\Lambda\alpha.\gamma(\delta(N))) \sigma \longrightarrow \gamma(\delta(N))[\alpha \mapsto \sigma] = \gamma(\delta_0(N))
\]
It suffices to show
\[
\gamma(\delta_0(N)) \sim_{\tau} \gamma'(\delta_0'(N)) [\eta_0 : \delta_0 \leftrightarrow \delta_0']
\]
which follows by IH from $\Delta, \alpha; \Gamma \vdash N : \tau$ (which we obtain from $\Delta, \Gamma \vdash \Lambda\alpha.N : \tau$ by inversion).
Proof of parametricity

Case $M$ is $N \sigma$:

By inversion of typing $\Delta, \Gamma \vdash N : \forall \alpha. \tau_0$ (1) and $\tau$ is $\forall \alpha. \tau_0$.

We must show that $\Delta; \Gamma \vdash N \sigma \sim N \sigma : \tau_0[\alpha \mapsto \sigma]$.

Assume $\eta : \delta \leftrightarrow \Delta \delta'$ and $\gamma \sim \Gamma \gamma' [\eta : \delta \leftrightarrow \delta']$. We must show

$$\gamma(\delta(N \sigma)) \sim_{\tau_0[\alpha \mapsto \sigma]} \gamma'(\delta'(N \sigma)) [\eta : \delta \leftrightarrow \delta']$$

i.e.

$$\gamma(\delta(N))) \sigma \sim_{\tau_0[\alpha \mapsto \sigma]} (\gamma'(\delta'(N))) \sigma [\eta : \delta \leftrightarrow \delta']$$

By compositionality, it suffices to show

$$\gamma(\delta(N))) \sigma \sim_{\tau_0} (\gamma'(\delta'(N))) \sigma [\eta_0 : \delta_0 \leftrightarrow \delta'_0] \quad (2)$$

where $\eta_0 = \eta, \alpha \mapsto R$ and $\delta_0 = \delta, \alpha \mapsto \sigma$ and $\delta'_0 = \delta, \alpha \mapsto \sigma'$ and $R : \delta(s) \leftrightarrow \delta'(s)$ is defined by $R(N_0, N'_0) \iff N_0 \sim_{\sigma} N'_0 [\eta : \delta \leftrightarrow \delta']$.

This relation is admissible (3). Hence by IH from (1), we have

$$(\gamma(\delta(N))) \sim_{\forall \alpha. \tau_0} (\gamma'(\delta'(N))) [\eta : \delta \leftrightarrow \delta']$$

which implies (2) by definition of $\sim_{\forall \alpha. \tau_0}$. 

$$\Box$$
Properties

**Theorem**

Logical equivalence and observational equivalence coincide. i.e. $\Delta; \Gamma \vdash M \sim M' : \tau$ iff $\Delta; \Gamma \vdash M \equiv M' : \tau$.

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

Extensionality

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

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

**Proof.** Forward direction is immediate as \( \cong \) is a congruence.

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

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

**Identity extension**

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

- Introduction
- Normalization of $\lambda_{st}$
- Observational equivalence in $\lambda_{st}$
- Logical relations in stlc
- Logical relations in F
- Applications
- Extensions
Tools

Value arguments with open types

\[ M \sim_{\tau_1 \to \tau_2} M' [\eta] \iff \forall V, V', (V \sim_{\tau_1} V' [\eta] \implies M V \sim_{\tau_2} M' V' [\eta]) \]

The implication follows from the definition.
The reverse is the value arguments lemma extended to open terms.
Hence, we could have used this as a definition.

Admissibility

A relation \( R \) is admissible iff it is the closure of a relation on values that is compatible with observational equivalence.
We may consider sets of \( \diamondsuit_{\tau \leftrightarrow \tau'} R \), admissible by construction, of the form

\[ \{(N, N') \mid \exists (M, M') \in R, \: N \equiv_{\tau} M \land M' \equiv_{\tau'} N'\} \]
Applications

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

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

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

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

Consider \( R \) equal to \( \triangleright_{\rho \leftrightarrow_{\rho}} (N, N) \) and \( \eta \) be \( \alpha \mapsto R : \rho \leftrightarrow_{\rho} \).

\( R \) is admissible by construction.

(Reminder: \( R(P, P') \) iff \( P \cong_{\rho} N \land N \cong_{\rho} P' \).)

Since \( R(N, N) \), we have \( N \sim_{\alpha} N [\eta] \) by definition.

Hence, from (2), we have \( M \rho N \sim_{\alpha} M \rho N [\eta] \), i.e. \( R(M \rho N, M \rho N) \), which implies (1) by definition of \( R \).
Applications

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

**Fact**  Let $\sigma$ be $\forall \alpha. \alpha \to \alpha \to \alpha$. If $M : \sigma$, then either

$$M \cong_\sigma M_1 \triangleq \Lambda \alpha. \lambda x_1 : \alpha. \lambda x_2 : \alpha. x_1$$

or

$$M \cong_\sigma M_2 \triangleq \Lambda \alpha. \lambda x_1 : \alpha. \lambda x_2 : \alpha. x_2$$

**Proof** By *extensionality*, it suffices to show that for either $i = 1$ or $i = 2$, for any closed type $\rho$ and $N_1, N_2 : \rho$, we have

$$M \rho N_1 N_2 \cong_\rho M_i \rho N_1 N_2,$$

or, by closure by inverse reduction and replacing observational by logical equivalence that $M \rho N_1 N_2 \cong_\sigma N_i \quad (1)$. Let $\rho$ and $N_1, N_2 : \rho$ be fixed.

Consider $R$ equal to $\diamond_{B \leftrightarrow \rho} \{(\text{tt}, N_1), (\text{ff}, N_2)\}$ and $\eta$ be $\alpha \mapsto R : B \leftrightarrow \rho$.

We have $\text{tt} \sim_\alpha N_1 \ [\eta]$ since $R(\text{tt}, N_1)$ and, similarly, $\text{ff} \sim_\alpha N_2 \ [\eta]$.

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

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

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

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

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

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

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

Let $\eta$ be $\alpha \mapsto R$ where $R : \mathbb{N} \leftrightarrow \rho$ is defined as $\triangleright_{\mathbb{N} \leftrightarrow \rho} \{(k, V_1^k V_2) \mid k \in \mathbb{N}\}$.

We have $0 \sim_{\alpha} V_2 \ [\eta]$ since $R(0, V_2)$ (reduce the right-hand side for $k = 0$).

We also have $\text{succ} \sim_{\alpha \to \alpha} V_1 \ [\eta]$. Indeed, assume $N \sim_{\alpha} N' \ [\eta]$, i.e. $R(N, N')$.

There exists $k$ such that $N \cong_{\mathbb{N}} k$ and $N' \cong_{\rho} V_1^k V_2$. By congruence, we have $\text{succ} N \cong_{\mathbb{N}} \text{succ} k \downarrow k + 1$ and $V_1 N' \cong_{\rho} V_1 (V_1^k V_2) \downarrow V_1^{k+1} V_2$. Hence $R(\text{succ} N, V_1 N')$, that is $\text{succ} N \sim_{\alpha} V_1 N' \ [\eta]$. 


Applications

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

**Fact** Let $\sigma$ be $\forall \alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$. If $M : \sigma$, then $M \equiv \sigma M_n$ for some integer $n$, where $M_n \Delta \equiv \lambda \alpha. \lambda f : \alpha \rightarrow \alpha. \lambda x : \alpha. f^n x$.

**Proof** By *extensionality*, and *value arguments*, it suffices to show that there exists $n$ such for any closed type $\rho$ and closed values $V_1 : \rho \rightarrow \rho$ and $V_2 : \rho$, we have $M \rho V_1 V_2 \equiv \rho M_n \rho V_1 V_2$, or, by closure by inverse reduction and replacing observational by logical equivalence, $M \rho V_1 V_2 \sim \rho V_1^n V_2$, (1).

Let $V_1 : \rho \rightarrow \rho$ and $V_2 : \rho$ be fixed.
Let $\eta$ be $\alpha \mapsto R$ where $R : \mathbb{N} \leftrightarrow \rho$ is defined as $\diamond_{\mathbb{N} \leftrightarrow \rho} \{ (k, V_1^k V_2) \mid k \in \mathbb{N} \}$.
We have $0 \sim_\alpha V_2 \ [\eta]$ since $R(0, V_2)$ (reduce the right-hand side for $k = 0$).

We also have $\text{succ} \sim_\alpha V_1 \ [\eta]$. (A key to the proof.)

By parametricity, we have $M \sim \sigma M$. Hence, $M \mathbb{N} \text{succ} 0 \sim_\alpha M \rho V_1 V_2 \ [\eta]$, i.e. $R(M \mathbb{N} \text{succ} 0, M \rho V_1 V_2)$ which means that there exists $n$ such that $M \mathbb{N} \text{succ} 0 \sim_\sigma n$ and $M \rho V_1 V_2 \sim_\rho V_1^n V_2$.

Since, $M \mathbb{N} \text{succ} 0$ is independent of $\rho$, $V_1$, and $V_2$, and all $n$ are in different equivalence classes at the base type $\mathbb{N}$, we may conclude (1).
Applications

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

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

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

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

Let $V_1 : \rho \to \rho$ and $V_2 : \rho$ be fixed. Let $Z$ and $S$ be $M_0 nat$ and $M_1 nat$.

Let $\eta$ be $\alpha \mapsto R$ where $R : nat \leftrightarrow \rho$ defined as $\Diamond_{nat \leftrightarrow \rho}^{R} \{(S^k Z, V_1^k V_2) \mid k \in \mathbb{N}\}$.

We have $Z \sim_{\alpha \to} V_2 \ [\eta]$ since $R(Z, V_2)$ (reduce both sides for $k = 0$).

We also have $S \sim_{\alpha \to \alpha} V_1 \ [\eta]$. 
Applications

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

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

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

Let $V_1 : \rho \to \rho$ and $V_2 : \rho$ be fixed. Let $Z$ and $S$ be $M_0 \text{nat}$ and $M_1 \text{nat}$.

Let $\eta$ be $\alpha \mapsto R$ where $R : \text{nat} \leftrightarrow \rho$ defined as $\Diamond^{R}_{\text{nat} \leftrightarrow \rho} \{(S^k Z, V_1^k V_2) \mid k \in \mathbb{N}\}$.

We have $Z \sim_{\alpha} V_2 [\eta]$ since $R(Z, V_2)$ (reduce both sides for $k = 0$).

We also have $S \sim_{\alpha \to \alpha} V_1 [\eta]$. 
Applications

**Inhabitants of** \( \forall \alpha. (\alpha \to \alpha) \to \alpha \to \alpha \)

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

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

Let \( V_1 : \rho \to \rho \) and \( V_2 : \rho \) be fixed. Let \( Z \) and \( S \) be \( M_0 \text{nat} \) and \( M_1 \text{nat} \).

Let \( \eta \) be \( \alpha \mapsto R \) where \( R : \text{nat} \leftrightarrow \rho \) defined as \( \diamond_{\text{nat} \leftrightarrow \rho} \{ (S^k Z, V_1^k V_2) \mid k \in \mathbb{N} \} \).

We have \( Z \sim_{\alpha} V_2 [\eta] \) since \( R(Z, V_2) \) (reduce both sides for \( k = 0 \)).

We also have \( S \sim_{\alpha \to \alpha} V_1 [\eta] \).
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 $M : nat$, then $M \cong_{nat} M_n$ for some integer $n$, where $M_n \triangleq \Lambda \alpha. \lambda f : \alpha \to \alpha. \lambda x : \alpha. f^n x$.

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

Let $V_1 : \rho \to \rho$ and $V_2 : \rho$ be fixed. Let $Z$ and $S$ be $M_0 nat$ and $M_1 nat$. Let $\eta$ be $\alpha \mapsto R$ where $R : nat \leftrightarrow \rho$ defined as $\diamond_{nat\leftrightarrow\rho}^{R} \{(S^kZ, V_1^kV_2) \mid k \in \mathbb{N}\}$. We have $Z \sim_{\alpha} V_2 [\eta]$ since $R(Z, V_2)$ (reduce both sides for $k = 0$).

We also have $S \sim_{\alpha \to \alpha} V_1 [\eta]$.

Indeed, assume $N \sim_{\alpha} N' [\eta]$, i.e. $R(N, N')$.

There exists $k$ such that $N \cong_{nat} S^kZ$ and $N' \cong_{\rho} V_1^kV_2$.

By congruence $SN \cong_{nat} S^{k+1}Z$ and $V_1 N' \cong_{\rho} V_1^{k+1}V_2$.

Therefore $R(SN, V_1 N')$, i.e. $SN \sim_{\alpha} V_1 N' [\eta]$. 
Applications

**Inhabitants of** \( \forall \alpha. (\alpha \to \alpha) \to \alpha \to \alpha \)

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

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

Let \( V_1 : \rho \to \rho \) and \( V_2 : \rho \) be fixed. Let \( Z \) and \( S \) be \( M_0 \text{ nat} \) and \( M_1 \text{ nat} \).
Let \( \eta \) be \( \alpha \mapsto R \) where \( R : \text{nat} \leftrightarrow \rho \) defined as \( \diamond_{\text{nat} \leftrightarrow \rho} (S^k Z, V_1^k V_2) \mid k \in \mathbb{N} \).

We have \( Z \sim_{\alpha} V_2 \ [\eta] \) since \( R(Z, V_2) \) (reduce both sides for \( k = 0 \)).

We also have \( S \sim_{\alpha \to \alpha} V_1 \ [\eta] \).

Indeed, assume \( N \sim_{\alpha} N' \ [\eta] \), i.e. \( R(N, N') \).
There exists \( k \) such that \( N \equiv_{\text{nat}} S^k Z \) and \( N' \equiv_{\rho} V_1^k V_2 \).
By congruence \( S N \equiv_{\text{nat}} S^{k+1} Z \) and \( V_1 N' \equiv_{\rho} V_1^{k+1} V_2 \).
Therefore \( R(S N, V_1 N') \), i.e. \( S N \sim_{\alpha} V_1 N' \ [\eta] \).
Applications

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

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

Let $V_1 : \rho \to \rho$ and $V_2 : \rho$ be fixed. Let $Z$ and $S$ be $M_0 \text{nat}$ and $M_1 \text{nat}$.

Let $\eta$ be $\alpha \mapsto R$ where $R : \text{nat} \leftrightarrow \rho$ defined as $\diamond_{\text{nat} \leftrightarrow \rho} \{(S^k Z, V_1^k V_2) \mid k \in \mathbb{N}\}$.

We have $Z \sim_{\alpha} V_2 \ [\eta]$ since $R(Z, V_2)$ (reduce both sides for $k = 0$).

We also have $S \sim_{\alpha \to \alpha} V_1 \ [\eta]$. (A key to the proof.)

By parametricity, we have $M \sim_{\text{nat}} M$. Hence, $M \text{nat} S Z \sim_{\alpha} M \rho V_1 V_2 \ [\eta]$, i.e. $R(M \text{nat} S Z, M \rho V_1 V_2)$ which means that there exists $n$ such that $M \text{nat} S Z \sim_{\text{nat}} S^n Z$ and $M \rho V_1 V_2 \sim_{\rho} V_1^n V_2$.

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

\[ \text{sort} : \forall \alpha. (\alpha \to \alpha \to \text{bool}) \to \text{list } \alpha \]

**Property** Assume \( \text{sort} : \forall \alpha. (\alpha \to \alpha \to \text{bool}) \to \text{list } \alpha \to \text{list } \alpha \) (2). Then

\[
(\forall x, y, \ cmp_2 (f \ x) (f \ y) = cmp_1 \ x \ y) \implies \\
\forall \ell, \ \text{sort } \ cmp_2 (\text{map } f \ \ell) = \text{map } f (\text{sort } cmp_1 \ \ell)
\]
Applications

\textit{sort} : \forall \alpha. (\alpha \to \alpha \to \text{bool}) \to \text{list} \alpha

**Proof** We have \( \text{sort} \sim_\sigma \text{sort} \) where \( \sigma \) is \( \forall \alpha. (\alpha \to \alpha \to \text{bool}) \to \text{list} \alpha \to \text{list} \alpha \).

By definition, for all \( \rho_1, \rho_2 \), and all admissible relations \( R : \delta_1 \leftrightarrow \delta_2 \), where \( \delta_i \) is \( \alpha \mapsto \rho_i \) for all \( cp_1, cp_2 \),

\[
\forall cp_1, cp_2, \quad cp_1 \sim_{\alpha \to \alpha \to \text{bool}} cp_2 [\eta] \quad \Longrightarrow \\
\forall V_1, V_2, \quad (V_1 \sim_{\text{list} \alpha} V_2 [\eta] \quad \Longrightarrow \quad \text{sort} \rho_1 \ cp_1 \ V_1 \sim_{\text{list} \alpha} \text{sort} \rho_2 \ cp_2 \ V_2 [\eta])
\]

(3)

We may restrict \( R(N_1, N_2) \), i.e. \( N_1 \sim_{\alpha} N_2 \) to \( N_2 = g \ N_1 \) for some function \( g \).

Then \( N_1 \sim_{\text{list} \alpha} N_2 [\eta] \) iff \( N_2 = \text{map} \ g \ N_1 \). Thus, (4) becomes

\[
\forall V_1 : \text{list} \ \rho, \quad \text{sort} \ \rho_2 \ cp_2 \ (\text{map} \ g \ V_1) = \text{map} \ g \ \text{(sort} \ \rho_1 \ cp_1 \ V_1)
\]

While (3) means

\[
\forall V_1, V_1', V_2, V_2', \quad V_1 \sim_{\alpha} V_2 [\eta] \land V_1' \sim_{\alpha} V_2' [\eta] \quad \Longrightarrow \quad cp_1 \ V_1 \ V_1' \sim_{B} cp_2 \ V_2 \ V_2' [\eta]
\]

i.e.

\[
\forall V, W : \rho, \quad cp_1 \ V \ W \sim_{B} cp_2 \ (g \ V) \ (g \ W)
\]
Encodable features

We have shown that all expressions of type \( \text{nat} \) behave as natural numbers. Hence, natural numbers are definable.

Still, we could also provide a type \( \text{nat} \) of natural numbers as primitive.

Then, if \( M : \text{nat} \) and \( M' : \text{nat} \), we have \( M \simeq_{\text{nat}} M' \) iff there exists \( V : \text{nat} \) such that \( M \downarrow V \) and \( M' \downarrow V \).

Then logical equivalence is defined as \( M \sim_{\text{nat}} M' \ [\eta] \) iff \( M \simeq_{\text{nat}} M' \)

All properties are preserved.
Encodable features

Given closed types $\tau_1$ and $\tau_2$, we defined

$$\tau_1 \times \tau_2 \triangleq \forall \alpha. (\tau_1 \to \tau_2 \to \alpha) \to \alpha$$

$$(M_1, M_2) \triangleq \Lambda \alpha. \lambda x : \tau_1 \to \tau_2 \to \alpha. x \ M_1 \ M_2$$

$$M . i \triangleq M \ (\lambda x_1 : \tau_1. \lambda x_2 : \tau_2. x_i)$$

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, we define:

$$M \sim_{\tau_1 \times \tau_2} M' \ [\eta : \delta \leftrightarrow \delta'] \iff \forall i \in \{1, 2\}, \ M . i \sim_{\tau_i} M' . i \ [\eta : \delta \leftrightarrow \delta']$$
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 $\sigma_1$ and $\sigma_2$.

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

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

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

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

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

- Introduction
- Normalization of $\lambda_{st}$
- Observational equivalence in $\lambda_{st}$
- Logical relations in stlc
- Logical relations in $F$
- Applications
- Extensions
Existential types

Definition?

\[
\text{pack } N_1, \rho_1 \text{ as } \exists \alpha. \tau \sim_{\exists \alpha. \tau} \text{ pack } N_2, \rho_2 \text{ as } \exists \alpha. \tau \quad [\eta : \delta_1 \leftrightarrow \delta_2]
\]

iff there exist \( R : \rho_1 \leftrightarrow \rho_2, \)

\[
N_1 \sim_{\tau} N_2 \quad [(\eta, \alpha \mapsto R) : (\delta_1, \alpha \mapsto \rho_1) \leftrightarrow (\delta_2, \alpha \mapsto \rho_2)]
\]

This definition is correct but incomplete as it only relates terms of existential types in head normal forms.
We may extend it to a relation between arbitrary terms by anti-reduction closure.
Alternative definition

Instead of defining the relation on terms, we may define the relation on values and lift it to a relation on terms by anti-reduction closure.

\[ \text{tt} \approx_B \text{tt} \ [\eta] \land \text{ff} \approx_B \text{ff} \ [\eta] \]

\[ \lambda x: \tau. M_1 \approx_{\tau \rightarrow \tau'} \lambda x: \tau. M_2 \ [\eta] \]

\[ \iff \forall V_1, V_2, V_1 \approx_{\tau} V_2 \ [\eta] \implies (\lambda x: \tau. M_1) V_1 \sim_{\tau_2} (\lambda x: \tau'. M_2) V_2 \ [\eta] \]

\[ V_1 \approx_\alpha V_2 \ [\eta] \]

\[ \iff \eta(\alpha)(V_1, V_2) \]

\[ \Lambda \alpha. V_1 \approx_{\forall \alpha. \tau} \Lambda \alpha. V_2 \ [\eta] \]

\[ \iff \forall \rho_1, \rho_2, R:\rho_1 \leftrightarrow \rho_2, (\Lambda \alpha. V_1) \rho_1 \sim_{\tau} (\Lambda \alpha. V_2) \rho_2 \ [\eta, \alpha \mapsto R] \]

\[ \textbf{pack} V_1, \rho_1 \textbf{ as } \exists \alpha. \tau \approx_{\exists \alpha. \tau} \textbf{pack} V_2, \rho_2 \textbf{ as } \exists \alpha. \tau \ [\eta] \]

\[ \iff \exists R:\rho_1 \leftrightarrow \rho_2, V_1 \approx_{\eta, \rho \mapsto R} V_2 \ [\eta, \alpha \mapsto R] \]

\[ M_1 \sim_{\tau} M_2 \ [\eta] \iff \exists V_1, V_2, M_1 \Downarrow V_1 \land M_2 \Downarrow V_2 \land V_1 \approx_{\tau} V_2 \ [\eta] \]

where \( R: \rho_1 \leftrightarrow \rho_2 \) means a relation between values of type \( \tau_1 \) and \( \tau_2 \).

(We may require compatibility with observational equivalence on values.)
Alternative definition (variant)

\[ \nu[\mathbf{B}]_\eta = \{(tt, tt), (ff, ff)\} \]

\[ \nu[\tau \rightarrow \tau']_\eta = \{(\lambda x: \tau. M_1, \lambda x: \tau. M_2) \mid \forall (V_1, V_2) \in \nu[\tau]_\eta, ((\lambda x: \tau. M_1) V_1, (\lambda x: \tau. M_2) V_2) \in \mathcal{E}[\tau']_\eta\} \]

\[ \nu[\forall \alpha. \tau]_\eta = \{(\Lambda \alpha. V_1, \Lambda \alpha. V_2) \mid \forall \rho_1, \rho_2, R: \rho_1 \leftrightarrow \rho_2, ((\Lambda \alpha. V_1) \rho_1, (\Lambda \alpha. V_2) \rho_2) \in \mathcal{E}[\tau]_\eta, \alpha \mapsto R\} \]

\[ \nu[\exists \alpha. \tau]_\eta = \{(\text{pack } V_1, \rho_1 \text{ as } \exists \alpha. \tau, \text{pack } V_2, \rho_2 \text{ as } \exists \alpha. \tau) \mid \exists R: \rho_1 \leftrightarrow \rho_2, (V_1, V_2) \in \nu[\tau]_\eta, \alpha \mapsto R\} \]

\[ \mathcal{E}[\tau]_\eta = \{(M_1, M_2) \mid \exists (V_1, V_2) \in \nu[\tau]_\eta, M_1 \downarrow V_1 \land M_2 \downarrow V_2\} \]
Existential types

Consider \( V_1 \triangleq (\text{not}, \text{tt}) \), and \( V_2 \triangleq (\text{succ}, 0) \) and \( \sigma \triangleq (\alpha \to \alpha) \times \alpha \).

Let \( R : \text{bool} \leftrightarrow \text{nat} \) be \( \{(\text{tt}, 2n), (\text{ff}, 2n + 1) \mid n \in \mathbb{N}\} \) and \( \eta \) be \( \alpha \mapsto R \).

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

Hence, \((\text{pack } V_1, \text{bool as } \exists \alpha \cdot \sigma, \text{pack } V_2, \text{nat as } \exists \alpha \cdot \sigma) \in \mathcal{V}[\exists \alpha \cdot \sigma]_\eta \).

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

We have \( (\text{tt}, 0) \in \mathcal{V}[\alpha]_{\eta} \), since \( (\text{tt}, 0) \in R \).

We also have \((\text{not}, \text{succ}) \in \mathcal{V}[\alpha \to \alpha]_{\eta} \) which proves (1).

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

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

In both cases, \( (\text{not } W_1, \text{succ } W_2) \) reduces to a pair in \( R \).

Hence, \((\text{not } W_1, \text{succ } W_2) \in \mathcal{E}[\alpha]_{\eta} \).
Extensions to products and sums

\[ \mathcal{V}[\tau \times \tau']_\eta = \{ (V_1, V_2) \mid V_1 \in \mathcal{V}[\tau]_\eta \land V_2 \in \mathcal{V}[\tau']_\eta \} \]

\[ \mathcal{V}[\tau + \tau']_\eta = \{ (\text{inj}_1 V_1, \text{inj}_1 V_2) \mid (V_1, V_2) \in \mathcal{V}[\tau]_\eta \} \cup \{ (\text{inj}_2 V_2, \text{inj}_2 V_2) \mid (V_1, V_2) \in \mathcal{V}[\tau']_\eta \} \]
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

\[ \mathcal{V}[\mu \alpha. \tau]_\eta = \mathcal{V}[\lbrack \alpha \mapsto \mu \alpha. \tau \rbrack \tau]_\eta \]

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 $\tau$ up to $n$ reduction steps.

\[
\mathcal{V}_k[\mathbb{B}]_\eta = \{(tt, tt), (ff, ff)\}
\]
\[
\mathcal{V}_k[\tau \rightarrow \tau']_\eta = \{(\lambda x : \tau. M_1, \lambda x : \tau. M_2) | \forall j < k, \forall (N_1, N_2) \in \mathcal{V}_j[\tau]_\eta, ((\lambda x : \tau. M_1) N_1, (\lambda x : \tau. M_2) N_2) \in \mathcal{E}_j[\tau']_\eta\}
\]
\[
\mathcal{V}_k[\alpha]_\eta = \eta(\alpha).k
\]
\[
\mathcal{V}_k[\forall \alpha. \tau]_\eta = \{(\Lambda \alpha. V_1, \Lambda \alpha. V_2) | \forall \rho_1, \rho_2, R \in \mathcal{R}^k(\rho_1, \rho_2), \forall j < k, ((\Lambda \alpha. V_1) \rho_1, (\Lambda \alpha. V_2) \rho_2) \in \mathcal{V}_j[\tau]_\eta, \rho \rightarrow R\}
\]
\[
\mathcal{V}_k[\mu \alpha. \tau]_\eta = \mathcal{V}_{k-1}[[\alpha \mapsto \mu \alpha. \tau]\tau]_\eta
\]
\[
\mathcal{E}_k[\tau]_\eta = \{(M_1, M_2) | \forall j < k, M_1 \downarrow_j V_1 \implies \exists V_2, M_2 \downarrow V_2 \land (V_1, V_2) \in \mathcal{V}_{k-j}[\tau]_\eta\}
\]

By $\downarrow_j$ means *reduces in $j$-steps*

$\mathcal{R}^j(\rho_1, \rho_2)$ is a sequence of decreasing relations between closed values of closed types $\rho_1$ and $\rho_2$ of length (at least) $j$. 

By $\downarrow_j$ means *reduces in $j$-steps*
Step-indexed logical relations (a taste)

The relation is asymmetric.

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

$$\forall \eta \in R^k_\Delta(\delta_1, \delta_2), \forall (\gamma_1, \gamma_2) \in G_k[\Gamma], \ (\gamma_1(\delta_1(M_1)), \gamma_2(\delta_2(M_2)) \in E_k[\tau]\eta$$

and

$$\Delta; \Gamma \vdash M_1 \sim M_2 : \tau \triangleq \bigwedge \left\{ \begin{array}{l} \Delta; \Gamma \vdash M_1 \preceq M_2 : \tau \\ \Delta; \Gamma \vdash M_2 \preceq M_1 : \tau \end{array} \right\}$$

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


