Fundamental Theorems for Free

Logical Relations & Parametricity for Substructural Type Systems and Beyond

Corinthia Beatrix Aberlé (she/her)

December 10, 2024

What is a Logical Relation?

Logical relations are a powerful proof technique that can be used to prove properties of type systems such as normalization, canonicity, and parametricity, that typically evade straightforward proof by induction.

What is a Logical Relation?

LOGICAL RELATIONS are a powerful proof technique that can be used to prove properties of type systems such as normalization, canonicity, and parametricity, that typically evade straightforward proof by induction.

But what is a logical relation?

What is a Logical Relation?

LOGICAL RELATIONS are a powerful proof technique that can be used to prove properties of type systems such as normalization, canonicity, and parametricity, that typically evade straightforward proof by induction.

But what is a logical relation?

The usual way of introducing logical relations is by example, with the term “logical relations” being applied to anything that sufficiently resembles other logical relations arguments, rather than having a precise definition.

What is a Logical Relation?

Logical relations are a powerful proof technique that can be used to prove properties of type systems such as normalization, canonicity, and parametricity, that typically evade straightforward proof by induction.

But what is a logical relation?

The usual way of introducing logical relations is by example, with the term "logical relations" being applied to anything that sufficiently resembles other logical relations arguments, rather than having a precise definition.

This lack of precision makes for difficulty in extending logical relations arguments to novel type systems, since there is not a clear standard by which to judge whether the relations one defines are the "right" ones.

What is a Logical Relation?

Logical relations are a powerful proof technique that can be used to prove properties of type systems such as normalization, canonicity, and parametricity, that typically evade straightforward proof by induction.

But what is a logical relation?

The usual way of introducing logical relations is by example, with the term "logical relations" being applied to anything that sufficiently resembles other logical relations arguments, rather than having a precise definition.

This lack of precision makes for difficulty in extending logical relations arguments to novel type systems, since there is not a clear standard by which to judge whether the relations one defines are the "right" ones.

In this talk, I will first sketch some of my own recent work on developing logical relations to prove parametricity theorems for substructural type systems, using this as a jumping-off point to discuss a more general *recipe* for logical relations, based on category theory, that can be used to derive these and other examples.

Recap: Simply-Typed λ-Calculus

Our starting point for most of the type systems considered in this talk will be the Simply-Typed λ-Calculus with both function and pair types:

$$\frac{}{\mathbf{1}\ \mathsf{Type}} \qquad \frac{A\ \mathsf{Type} \qquad B\ \mathsf{Type}}{A \times B\ \mathsf{Type}} \qquad \frac{A\ \mathsf{Type} \qquad B\ \mathsf{Type}}{A \longrightarrow B\ \mathsf{Type}}$$

$$\frac{}{\Gamma, x : A, \Gamma' \vdash x : A} \qquad \frac{}{\Gamma \vdash () : \mathbf{1}}$$

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash (a, b) : A \times B} \qquad \frac{\Gamma \vdash p : A \times B}{\Gamma \vdash \pi_1(p) : A} \qquad \frac{\Gamma \vdash p : A \times B}{\Gamma \vdash \pi_2(p) : B}$$

$$\frac{\Gamma, x : A \vdash f : B}{\Gamma \vdash \lambda x.f : A \longrightarrow B} \qquad \frac{\Gamma \vdash f : A \longrightarrow B \qquad \Gamma \vdash a : A}{\Gamma \vdash f(a) : B}$$

$$\pi_1(a, b) \equiv_A a \qquad \pi_2(a, b) \equiv_B b \qquad p \equiv_{A \times B} (\pi_1(p), \pi_2(p))$$

$$(\lambda x.f)(a) \equiv_B f[a/x] \qquad f \equiv_{A \to B} \lambda x.f(x) \qquad u \equiv_{\mathbf{1}} ()$$

Example 1: System T

System T extends simply-typed λ-calculus with a type of natural numbers:

$$\frac{}{\mathbb{N}\ \mathsf{Type}} \qquad \frac{}{\Gamma \vdash 0 : \mathbb{N}} \qquad \frac{\Gamma \vdash n : \mathbb{N}}{\Gamma \vdash \mathbf{s}(n) : \mathbb{N}}$$

$$\frac{\Gamma \vdash n : \mathbb{N} \qquad \Gamma \vdash a_0 : \mathrm{A} \qquad \Gamma, x : \mathbb{N}, y : \mathrm{A} \vdash a_1 : \mathrm{A}}{\Gamma \vdash \mathsf{rec}\ n\ \{0 \mapsto a_0 \mid \mathbf{s}(x), y \mapsto a_1\} : \mathrm{A}}$$

$$\mathsf{rec}\ 0\ \{0 \mapsto a_0 \mid \dots\} \equiv_{\mathrm{A}} a_0$$

$$\mathsf{rec}\ \mathbf{s}(n)\ \{\cdots \mid \mathbf{s}(x), y \mapsto a_1\} \equiv_{\mathrm{A}} a_1[n/x, \mathsf{rec}\ n\ \{\dots\}/y]$$

Example 1: System T

System T extends simply-typed λ-calculus with a type of natural numbers:

$$\frac{}{\mathbb{N}\ \mathsf{Type}} \qquad \frac{}{\Gamma \vdash 0 : \mathbb{N}} \qquad \frac{\Gamma \vdash n : \mathbb{N}}{\Gamma \vdash \mathbf{s}(n) : \mathbb{N}}$$

$$\frac{\Gamma \vdash n : \mathbb{N} \qquad \Gamma \vdash a_0 : \mathrm{A} \qquad \Gamma, x : \mathbb{N}, y : \mathrm{A} \vdash a_1 : \mathrm{A}}{\Gamma \vdash \mathsf{rec}\ n\ \{0 \mapsto a_0 \mid \mathbf{s}(x), y \mapsto a_1\} : \mathrm{A}}$$

$$\mathsf{rec}\ 0\ \{0 \mapsto a_0 \mid \dots\} \equiv_{\mathrm{A}} a_0$$

$$\mathsf{rec}\ \mathbf{s}(n)\ \{\cdots \mid \mathbf{s}(x), y \mapsto a_1\} \equiv_{\mathrm{A}} a_1[n/x, \mathsf{rec}\ n\ \{\dots\}/y]$$

For any natural number m, let $\overline{m} : \mathbb{N}$ be defined inductively as follows:

$$\begin{aligned} \overline{0} &= 0 \\ \overline{m+1} &= \mathbf{s}(\overline{m}) \end{aligned}$$

Canonicity for System T

Theorem (Canonicity): for every closed term $n : \mathbb{N}$, there exists a natural number m such that $n \equiv_{\mathbb{N}} \overline{m}$.

Canonicity for System T

Theorem (Canonicity): for every closed term $n : \mathbb{N}$, there exists a natural number m such that $n \equiv_{\mathbb{N}} \overline{m}$.

For each τ Type in System T, let $[\![\tau]\!]$ be the set of closed terms of type τ, quotiented up to definitional equality $\equiv_\tau$.

Canonicity for System T

Theorem (Canonicity): for every closed term $n : \mathbb{N}$, there exists a natural number m such that $n \equiv_{\mathbb{N}} \overline{m}$.

For each τ Type in System T, let $[\![\tau]\!]$ be the set of closed terms of type τ, quotiented up to definitional equality $\equiv_\tau$.

Then for each τ Type, define a predicate $\mathbb{P}_\tau \subseteq [\![\tau]\!]$, as follows:

$$
\begin{array}{rcl}
\mathbb{P}_{\mathbb{N}}(n) & \iff & \exists m \text{ s.t. } n \equiv_{\mathbb{N}} \overline{m} \\
\mathbb{P}_{A \times B}(p) & \iff & \mathbb{P}_A(\pi_1(p)) \text{ and } \mathbb{P}_B(\pi_2(p)) \\
\mathbb{P}_{A \to B}(f) & \iff & \forall a \in [\![A]\!].\mathbb{P}_A(a) \implies \mathbb{P}_B(f(a))
\end{array}
$$

Canonicity for System T

Theorem (Canonicity): for every closed term $n : \mathbb{N}$, there exists a natural number m such that $n \equiv_{\mathbb{N}} \overline{m}$.

For each τ Type in System T, let $[\![\tau]\!]$ be the set of closed terms of type τ, quotiented up to definitional equality $\equiv_\tau$.

Then for each τ Type, define a predicate $\mathbb{P}_\tau \subseteq [\![\tau]\!]$, as follows:

$$
\begin{aligned}
\mathbb{P}_{\mathbb{N}}(n) &\iff \exists m \text{ s.t. } n \equiv_{\mathbb{N}} \overline{m} \\
\mathbb{P}_{A\times B}(p) &\iff \mathbb{P}_A(\pi_1(p)) \text{ and } \mathbb{P}_B(\pi_2(p)) \\
\mathbb{P}_{A\to B}(f) &\iff \forall a \in [\![A]\!].\mathbb{P}_A(a) \implies \mathbb{P}_B(f(a))
\end{aligned}
$$

$[\![-]\!]$ and $\mathbb{P}$ straightforwardly lift to contexts Γ in addition to types τ.

Canonicity for System T

Theorem (Canonicity): for every closed term $n : \mathbb{N}$, there exists a natural number m such that $n \equiv_{\mathbb{N}} \overline{m}$.

For each τ Type in System T, let $[\![\tau]\!]$ be the set of closed terms of type τ, quotiented up to definitional equality $\equiv_\tau$.

Then for each τ Type, define a predicate $\mathbb{P}_\tau \subseteq [\![\tau]\!]$, as follows:

$$\begin{aligned}
\mathbb{P}_{\mathbb{N}}(n) &\iff \exists m \text{ s.t. } n \equiv_{\mathbb{N}} \overline{m} \\
\mathbb{P}_{A\times B}(p) &\iff \mathbb{P}_A(\pi_1(p)) \text{ and } \mathbb{P}_B(\pi_2(p)) \\
\mathbb{P}_{A\to B}(f) &\iff \forall a \in [\![A]\!].\mathbb{P}_A(a) \implies \mathbb{P}_B(f(a))
\end{aligned}$$

$[\![-]\!]$ and $\mathbb{P}$ straightforwardly lift to contexts Γ in addition to types τ.

Fundamental Theorem (FTLR): for every $\Gamma \vdash a : A$ and $\gamma : \Gamma$

$$\text{if } \mathbb{P}_\Gamma(\gamma) \text{ then } \mathbb{P}_A(a[\gamma/\Gamma])$$

Canonicity for System T

Theorem (Canonicity): for every closed term $n : \mathbb{N}$, there exists a natural number m such that $n \equiv_{\mathbb{N}} \overline{m}$.

For each τ Type in System T, let $[\![\tau]\!]$ be the set of closed terms of type τ, quotiented up to definitional equality $\equiv_\tau$.

Then for each τ Type, define a predicate $\mathbb{P}_\tau \subseteq [\![\tau]\!]$, as follows:

$$\begin{aligned}
\mathbb{P}_{\mathbb{N}}(n) &\iff \exists m \text{ s.t. } n \equiv_{\mathbb{N}} \overline{m} \\
\mathbb{P}_{A\times B}(p) &\iff \mathbb{P}_A(\pi_1(p)) \text{ and } \mathbb{P}_B(\pi_2(p)) \\
\mathbb{P}_{A\to B}(f) &\iff \forall a \in [\![A]\!].\mathbb{P}_A(a) \implies \mathbb{P}_B(f(a))
\end{aligned}$$

$[\![-]\!]$ and $\mathbb{P}$ straightforwardly lift to contexts Γ in addition to types τ.

Fundamental Theorem (FTLR): for every $\Gamma \vdash a : A$ and $\gamma : \Gamma$

$$\text{if } \mathbb{P}_\Gamma(\gamma) \text{ then } \mathbb{P}_A(a[\gamma/\Gamma])$$

Proof: induction on derivations.

Canonicity for System T

Theorem (Canonicity): for every closed term $n : \mathbb{N}$, there exists a natural number m such that $n \equiv_{\mathbb{N}} \overline{m}$.

For each τ Type in System T, let $[\![\tau]\!]$ be the set of closed terms of type τ, quotiented up to definitional equality $\equiv_\tau$.

Then for each τ Type, define a predicate $\mathbb{P}_\tau \subseteq [\![\tau]\!]$, as follows:

$$
\begin{aligned}
\mathbb{P}_{\mathbb{N}}(n) &\iff \exists m \text{ s.t. } n \equiv_{\mathbb{N}} \overline{m} \\
\mathbb{P}_{A \times B}(p) &\iff \mathbb{P}_A(\pi_1(p)) \text{ and } \mathbb{P}_B(\pi_2(p)) \\
\mathbb{P}_{A \to B}(f) &\iff \forall a \in [\![A]\!].\mathbb{P}_A(a) \implies \mathbb{P}_B(f(a))
\end{aligned}
$$

$[\![-]\!]$ and $\mathbb{P}$ straightforwardly lift to contexts Γ in addition to types τ.

Fundamental Theorem (FTLR): for every $\Gamma \vdash a : A$ and $\gamma : \Gamma$

$$\text{if } \mathbb{P}_\Gamma(\gamma) \text{ then } \mathbb{P}_A(a[\gamma/\Gamma])$$

Proof: induction on derivations.

Canonicity then follows as a corollary of the fundamental theorem.

Example 2: Polymorphism & System F

System F extends simply-typed λ-calculus with parametric polymorphism:

$$\frac{}{\Delta, X, \Delta' \mid X\ \mathsf{Type}} \qquad \frac{\Delta, X \mid A\ \mathsf{Type}}{\Delta \mid \forall X.A\ \mathsf{Type}}$$

$$\frac{\Delta, X \mid \Gamma \vdash F : A}{\Delta \mid \Gamma \vdash \Lambda X.F : \forall X.A} \qquad \frac{\Delta \mid \Gamma \vdash F : \forall X : A \qquad \Delta \mid B\ \mathsf{Type}}{\Delta \mid \Gamma \vdash F[B] : A[B/X]}$$

Example 2: Polymorphism & System F

System F extends simply-typed λ-calculus with parametric polymorphism:

$$\frac{}{\Delta, X, \Delta' \mid X\ \mathsf{Type}} \qquad \frac{\Delta, X \mid A\ \mathsf{Type}}{\Delta \mid \forall X.A\ \mathsf{Type}}$$

$$\frac{\Delta, X \mid \Gamma \vdash F : A}{\Delta \mid \Gamma \vdash \Lambda X.F : \forall X.A} \qquad \frac{\Delta \mid \Gamma \vdash F : \forall X : A \qquad \Delta \mid B\ \mathsf{Type}}{\Delta \mid \Gamma \vdash F[B] : A[B/X]}$$

Intuitively, Polymorphic functions in System F can't inspect the types over which they are defined and so must behave uniformly for all types at which they are instantiated. But how to make this idea precise?

Example 2: Polymorphism & System F

System F extends simply-typed λ-calculus with parametric polymorphism:

$$\frac{}{\Delta, X, \Delta' \mid X\ \mathsf{Type}} \qquad \frac{\Delta, X \mid A\ \mathsf{Type}}{\Delta \mid \forall X.A\ \mathsf{Type}}$$

$$\frac{\Delta, X \mid \Gamma \vdash F : A}{\Delta \mid \Gamma \vdash \Lambda X.F : \forall X.A} \qquad \frac{\Delta \mid \Gamma \vdash F : \forall X : A \qquad \Delta \mid B\ \mathsf{Type}}{\Delta \mid \Gamma \vdash F[B] : A[B/X]}$$

Intuitively, Polymorphic functions in System F can't inspect the types over which they are defined and so must behave uniformly for all types at which they are instantiated. But how to make this idea precise?

Reynolds (1983): Polymorphic functions should preserve all predicates/relations definable on closed types.

Example 2: Polymorphism & System F

System F extends simply-typed λ-calculus with parametric polymorphism:

$$\frac{}{\Delta, X, \Delta' \mid X \text{ Type}} \qquad \frac{\Delta, X \mid A \text{ Type}}{\Delta \mid \forall X.A \text{ Type}}$$

$$\frac{\Delta, X \mid \Gamma \vdash F : A}{\Delta \mid \Gamma \vdash \Lambda X.F : \forall X.A} \qquad \frac{\Delta \mid \Gamma \vdash F : \forall X : A \qquad \Delta \mid B \text{ Type}}{\Delta \mid \Gamma \vdash F[B] : A[B/X]}$$

Intuitively, Polymorphic functions in System F can't inspect the types over which they are defined and so must behave uniformly for all types at which they are instantiated. But how to make this idea precise?

Reynolds (1983): Polymorphic functions should preserve all predicates/relations definable on closed types.

This idea is then made precise using a logical relations construction.

(Unary) Parametricity for System F

For each System F type $X_1, \ldots, X_n \mid A\ \mathsf{Type}$, given types $B_1, \ldots, B_n$ and predicates $P_i \subseteq [\![B_i]\!]$ for $i = 1, \ldots, n$, define a predicate

$$\mathbb{P}_A^{X_i \mapsto P_i} \subseteq [\![A[B_1/X_1, \ldots, B_n/X_n]]\!]$$

(Unary) Parametricity for System F

For each System F type $X_1, \ldots, X_n \mid A$ Type, given types $B_1, \ldots, B_n$ and predicates $P_i \subseteq [\![B_i]\!]$ for $i = 1, \ldots, n$, define a predicate

$$\mathbb{P}_A^{X_i \mapsto P_i} \subseteq [\![A[B_1/X_1, \ldots, B_n/X_n]]\!]$$

as follows:

$$\begin{aligned}
\mathbb{P}_{X_j}^{X_i \mapsto P_i}(x) &\iff P_j(x) \\
\mathbb{P}_{\mathbf{1}}^{X_i \mapsto P_i}(u) &\iff u \equiv_{\mathbf{1}} () \\
\mathbb{P}_{A \times B}^{X_i \mapsto P_i}(p) &\iff \mathbb{P}_A^{X_i \mapsto P_i}(\pi_1(p)) \text{ and } \mathbb{P}_B^{X_i \mapsto P_i}(\pi_2(p)) \\
\mathbb{P}_{A \to B}^{X_i \mapsto P_i}(f) &\iff \forall a : A.\ \mathbb{P}_A^{X_i \mapsto P_i}(a) \implies \mathbb{P}_B^{X_i \mapsto P_i}(f(a)) \\
\mathbb{P}_{\forall X.A}^{X_i \mapsto P_i}(F) &\iff \forall B \text{ Type},\ P \subseteq [\![B]\!].\ \mathbb{P}_A^{X_i \mapsto P_i,\ X \mapsto P}(F[B])
\end{aligned}$$

(Unary) Parametricity for System F

For each System F type $X_1, \ldots, X_n \mid A$ Type, given types $B_1, \ldots, B_n$ and predicates $P_i \subseteq [\![B_i]\!]$ for $i = 1, \ldots, n$, define a predicate

$$\mathbb{P}_A^{X_i \mapsto P_i} \subseteq [\![A[B_1/X_1, \ldots, B_n/X_n]]\!]$$

as follows:

$$\begin{aligned}
\mathbb{P}_{X_j}^{X_i \mapsto P_i}(x) &\iff P_j(x) \\
\mathbb{P}_{\mathbf{1}}^{X_i \mapsto P_i}(u) &\iff u \equiv_{\mathbf{1}} () \\
\mathbb{P}_{A \times B}^{X_i \mapsto P_i}(p) &\iff \mathbb{P}_A^{X_i \mapsto P_i}(\pi_1(p)) \text{ and } \mathbb{P}_B^{X_i \mapsto P_i}(\pi_2(p)) \\
\mathbb{P}_{A \to B}^{X_i \mapsto P_i}(f) &\iff \forall a : A.\ \mathbb{P}_A^{X_i \mapsto P_i}(a) \implies \mathbb{P}_B^{X_i \mapsto P_i}(f(a)) \\
\mathbb{P}_{\forall X.A}^{X_i \mapsto P_i}(F) &\iff \forall B \text{ Type},\ P \subseteq [\![B]\!].\ \mathbb{P}_A^{X_i \mapsto P_i,\ X \mapsto P}(F[B])
\end{aligned}$$

FTLR: for all $X_1, \ldots, X_n \mid \Gamma \vdash a : A$, given closed types $B_1, \ldots, B_n$ with predicates $P_1, \ldots, P_n$ as above, and $\gamma : \Gamma$

$$\text{if } \mathbb{P}_\Gamma^{X_i \mapsto P_i}(\gamma) \text{ then } \mathbb{P}_A^{X_i \mapsto P_i}(a[\gamma/\Gamma])$$

Consequences of Parametricity, part 1

The Old Chestnut: every closed term $\alpha : \forall X.X \to X$ is extensionally equivalent to the polymorphic identity function $\Lambda X.\lambda x.x$.

Consequences of Parametricity, part 1

The Old Chestnut: every closed term $\alpha : \forall X.X \to X$ is extensionally equivalent to the polymorphic identity function $\Lambda X.\lambda x.x$.

Proof:

Consequences of Parametricity, part 1

The Old Chestnut: every closed term $\alpha : \forall X.X \rightarrow X$ is extensionally equivalent to the polymorphic identity function $\Lambda X.\lambda x.x$.

Proof:

- By parametricity, we know $\mathbb{P}_{\forall X.X \rightarrow X}(\alpha)$.

Consequences of Parametricity, part 1

The Old Chestnut: every closed term $\alpha : \forall X.X \to X$ is extensionally equivalent to the polymorphic identity function $\Lambda X.\lambda x.x$.

Proof:

- By parametricity, we know $\mathbb{P}_{\forall X.X \to X}(\alpha)$.
- So $\mathbb{P}^{X \mapsto P}_{X \to X}(\alpha[A])$ for all closed types A Type and $P \subseteq [\![A]\!]$.

Consequences of Parametricity, part 1

The Old Chestnut: every closed term $\alpha : \forall X.X \to X$ is extensionally equivalent to the polymorphic identity function $\Lambda X.\lambda x.x$.

Proof:

- By parametricity, we know $\mathbb{P}_{\forall X.X \to X}(\alpha)$.
- So $\mathbb{P}^{X \mapsto P}_{X \to X}(\alpha[A])$ for all closed types A Type and $P \subseteq [\![A]\!]$.
- Therefore $\mathbb{P}^{X \mapsto P}_{X}(\alpha[A](a))$ – i.e. $P(\alpha[A](a))$ – for all $a : A$ such that $P(a)$.

Consequences of Parametricity, part 1

The Old Chestnut: every closed term $\alpha : \forall X.X \to X$ is extensionally equivalent to the polymorphic identity function $\Lambda X.\lambda x.x$.

Proof:

- By parametricity, we know $\mathbb{P}_{\forall X.X \to X}(\alpha)$.
- So $\mathbb{P}^{X \mapsto P}_{X \to X}(\alpha[A])$ for all closed types A Type and $P \subseteq [\![A]\!]$.
- Therefore $\mathbb{P}^{X \mapsto P}_{X}(\alpha[A](a))$ – i.e. $P(\alpha[A](a))$ – for all $a : A$ such that $P(a)$.
- Hence for any closed type A and $a : A$, we can define $P \subseteq [\![A]\!]$ by $P = \{a\}$. By construction $b \in P \iff b \equiv_A a$, and so by the above it follows that $\alpha[A](a) \equiv_A a$.

Consequences of Parametricity, part 2

Further Example: every closed term $\alpha : \forall X.X \to X \to X \times X$ is extensionally equivalent to one of the following four functions:

$$\Lambda X.\lambda x.\lambda y.(x, y) \quad \Lambda X.\lambda x.\lambda y.(y, x) \quad \Lambda X.\lambda x.\lambda y.(x, x) \quad \Lambda X.\lambda x.\lambda y.(y, y)$$

Consequences of Parametricity, part 2

Further Example: every closed term $\alpha : \forall X. X \to X \to X \times X$ is extensionally equivalent to one of the following four functions:

$$\Lambda X.\lambda x.\lambda y.(x, y) \quad \Lambda X.\lambda x.\lambda y.(y, x) \quad \Lambda X.\lambda x.\lambda y.(x, x) \quad \Lambda X.\lambda x.\lambda y.(y, y)$$

Proof:

Consequences of Parametricity, part 2

Further Example: every closed term $\alpha : \forall X.X \to X \to X \times X$ is extensionally equivalent to one of the following four functions:

$$\Lambda X.\lambda x.\lambda y.(x, y) \quad \Lambda X.\lambda x.\lambda y.(y, x) \quad \Lambda X.\lambda x.\lambda y.(x, x) \quad \Lambda X.\lambda x.\lambda y.(y, y)$$

Proof:

- As before, we can unfold the parametricity theorem for $\alpha : \forall X.X \to X \to X \times X$ to the following:

$$\begin{array}{l} \forall A\ \mathsf{Type},\ P \subseteq [\![A]\!],\ a_0, a_1 : A.\ P(a_0) \text{ and } P(a_1) \\ \quad \Longrightarrow\ P(\pi_1(\alpha[A](a_0)(a_1))) \text{ and } P(\pi_2(\alpha[A](a_0)(a_1))) \end{array}$$

Consequences of Parametricity, part 2

Further Example: every closed term $\alpha : \forall X.X \to X \to X \times X$ is extensionally equivalent to one of the following four functions:

$$\Lambda X.\lambda x.\lambda y.(x, y) \quad \Lambda X.\lambda x.\lambda y.(y, x) \quad \Lambda X.\lambda x.\lambda y.(x, x) \quad \Lambda X.\lambda x.\lambda y.(y, y)$$

Proof:

- As before, we can unfold the parametricity theorem for $\alpha : \forall X.X \to X \to X \times X$ to the following:

$$\begin{aligned} &\forall A \text{ Type},\ P \subseteq [\![A]\!],\ a_0, a_1 : A.\ P(a_0) \text{ and } P(a_1) \\ &\quad \Longrightarrow\ P(\pi_1(\alpha[A](a_0)(a_1))) \text{ and } P(\pi_2(\alpha[A](a_0)(a_1))) \end{aligned}$$

- Hence for any A Type with $a_0, a_1 : A$, we can take $P = \{a_0, a_1\}$, by which it follows that

$$\pi_1(\alpha[A](a_0)(a_1)) \in \{a_0, a_1\} \quad \text{and} \quad \pi_2(\alpha[A](a_0)(a_1)) \in \{a_0, a_1\}$$

Ordered STLC

To make STLC reflect the rules of ordered logic, we first modify the variable rule so that a variable must be the only variable in context when it is used:

$$\frac{}{x : \mathrm{A} \vdash x : \mathrm{A}}$$

Ordered STLC

To make STLC reflect the rules of ordered logic, we first modify the variable rule so that a variable must be the only variable in context when it is used:

$$\frac{}{x : \mathrm{A} \vdash x : \mathrm{A}}$$

We then have the following modified types and rules, which now must preserve the relative order and multiplicities of variables in contexts:

$$\frac{}{\mathbf{1}\ \mathsf{Type}} \qquad \frac{\mathrm{A}\ \mathsf{Type} \qquad \mathrm{B}\ \mathsf{Type}}{\mathrm{A} \otimes \mathrm{B}\ \mathsf{Type}} \qquad \frac{\mathrm{A}\ \mathsf{Type} \qquad \mathrm{B}\ \mathsf{Type}}{\mathrm{A}/\mathrm{B}\ \mathsf{Type}} \qquad \frac{\mathrm{A}\ \mathsf{Type} \qquad \mathrm{B}\ \mathsf{Type}}{\mathrm{A} \setminus \mathrm{B}\ \mathsf{Type}}$$

$$\frac{}{\vdash \langle\rangle : \mathbf{1}} \qquad \frac{\Delta \vdash u : \mathbf{1} \qquad \Gamma, \Theta \vdash c : \mathrm{C}}{\Gamma, \Delta, \Theta \vdash \mathsf{let}\ \langle\rangle = u\ \mathsf{in}\ c : \mathrm{C}}$$

$$\frac{\Gamma \vdash a : \mathrm{A} \qquad \Delta \vdash b : \mathrm{B}}{\Gamma, \Delta \vdash \langle a, b\rangle : \mathrm{A} \otimes \mathrm{B}} \qquad \frac{\Delta \vdash p : \mathrm{A} \otimes \mathrm{B} \qquad \Gamma, x : \mathrm{A}, y : \mathrm{B}, \Theta \vdash c : \mathrm{C}}{\Gamma, \Delta, \Theta \vdash \mathsf{let}\ \langle x, y\rangle = p\ \mathsf{in}\ c : \mathrm{C}}$$

$$\frac{x : \mathrm{A}, \Gamma \vdash f : \mathrm{B}}{\Gamma \vdash \lambda x.f : \mathrm{A}/\mathrm{B}} \qquad \frac{\Gamma \vdash a : \mathrm{A} \qquad \Delta \vdash f : \mathrm{A}/\mathrm{B}}{\Gamma, \Delta \vdash f(a) : \mathrm{B}} \qquad \frac{\Gamma, x : \mathrm{A} \vdash f : \mathrm{B}}{\Gamma \vdash \lambda x.f : \mathrm{B} \setminus \mathrm{A}} \qquad \frac{\Gamma \vdash f : \mathrm{B} \setminus \mathrm{A} \qquad \Delta \vdash a : \mathrm{A}}{\Gamma, \Delta \vdash f(a) : \mathrm{B}}$$

Substructural Logic & Resources

A common interpretation of substructural logic & type theory is as a logic of *resources*, where the rules of the logic/type theory reflect the ways in which resources may be created, transformed, and consumed.

Substructural Logic & Resources

A common interpretation of substructural logic & type theory is as a logic of *resources*, where the rules of the logic/type theory reflect the ways in which resources may be created, transformed, and consumed.

We can represent such resources as a *monoid*, i.e. a set M equipped with an associative binary operation $\otimes : \mathrm{M} \times \mathrm{M} \longrightarrow \mathrm{M}$ and a unit element $\epsilon \in \mathrm{M}$, i.e. subject to the following equations

$$\epsilon \otimes m = m = m \otimes \epsilon \qquad (k \otimes m) \otimes n = k \otimes (m \otimes n) \qquad \forall k, m, n \in \mathrm{M}$$

Substructural Logic & Resources

A common interpretation of substructural logic & type theory is as a logic of *resources*, where the rules of the logic/type theory reflect the ways in which resources may be created, transformed, and consumed.

We can represent such resources as a *monoid*, i.e. a set M equipped with an associative binary operation $\otimes : \mathrm{M} \times \mathrm{M} \longrightarrow \mathrm{M}$ and a unit element $\epsilon \in \mathrm{M}$, i.e. subject to the following equations

$$\epsilon \otimes m = m = m \otimes \epsilon \qquad (k \otimes m) \otimes n = k \otimes (m \otimes n) \qquad \forall k, m, n \in \mathrm{M}$$

Intuitively, $\otimes$ allows us to *accumulate* resources, and ϵ represents *no resource*.

Substructural Logical Relations

Fix a monoid M. For each type A Type in substructural STLC, rather than a mere predicate $P \subseteq [\![A]\!]$, we define a relation $\Vdash_A \subseteq M \times [\![A]\!]$ – where $m \Vdash_A a$ roughly means that a satisfies the logical predicate in the presence of m resources – inductively as follows:

Substructural Logical Relations

Fix a monoid M. For each type A Type in substructural STLC, rather than a mere predicate $P \subseteq [\![A]\!]$, we define a relation $\Vdash_A \subseteq M \times [\![A]\!]$ – where $m \Vdash_A a$ roughly means that a satisfies the logical predicate in the presence of m resources – inductively as follows:

$$
\begin{aligned}
m \Vdash_{\mathbf{1}} u \quad &\Longleftrightarrow \quad m = \epsilon \text{ and } u \equiv_{\mathbf{1}} \langle\rangle \\
m \Vdash_{A \otimes B} p \quad &\Longleftrightarrow \quad \exists n, k \in M,\ a \in [\![A]\!],\ b \in [\![B]\!] \text{ such that} \\
&\qquad m = n \otimes k,\ p \equiv_{A \otimes B} \langle a, b \rangle,\ n \Vdash_A a,\ \text{and } k \Vdash_B b \\
m \Vdash_{A/B} f \quad &\Longleftrightarrow \quad \forall n \in M,\ a \in [\![A]\!],\ n \Vdash_A a \implies n \otimes m \Vdash_B f(a) \\
m \Vdash_{B \backslash A} f \quad &\Longleftrightarrow \quad \forall n \in M,\ a \in [\![A]\!],\ n \Vdash_A a \implies m \otimes n \Vdash_B f(a)
\end{aligned}
$$

Substructural Parametricity

Adding parametric polymorphism to substructural STLC works exactly the same as for System F.

$$\frac{}{\Delta, X, \Delta' \mid X \; \mathsf{Type}} \qquad \frac{\Delta, X \mid A \; \mathsf{Type}}{\Delta \mid \forall X.A \; \mathsf{Type}}$$

$$\frac{\Delta, X \mid \Gamma \vdash F : A}{\Delta \mid \Gamma \vdash \Lambda X.F : \forall X.A} \qquad \frac{\Delta \mid \Gamma \vdash F : \forall X : A \qquad \Delta \mid B \; \mathsf{Type}}{\Delta \mid \Gamma \vdash F[B] : A[B/X]}$$

Substructural Parametricity

Adding parametric polymorphism to substructural STLC works exactly the same as for System F.

$$\frac{}{\Delta, X, \Delta' \mid X\ \mathsf{Type}} \qquad \frac{\Delta, X \mid A\ \mathsf{Type}}{\Delta \mid \forall X.A\ \mathsf{Type}}$$

$$\frac{\Delta, X \mid \Gamma \vdash F : A}{\Delta \mid \Gamma \vdash \Lambda X.F : \forall X.A} \qquad \frac{\Delta \mid \Gamma \vdash F : \forall X : A \qquad \Delta \mid B\ \mathsf{Type}}{\Delta \mid \Gamma \vdash F[B] : A[B/X]}$$

But now we define parametricity for these types with respect to all relations $R \subseteq M \times [\![A]\!]$, rather than just predicates on closed terms.

$$m \Vdash_{X_j}^{X_i \mapsto R_i} x \iff m\ R_j\ x$$

$$m \Vdash_{\forall X.A}^{X_i \mapsto R_i} F \iff \forall B\ \mathsf{Type},\ R \subseteq M \times [\![B]\!],\ m \Vdash_{A}^{X_i \mapsto R_i,\ X \mapsto R} F[B]$$

The Fundamental Theorem

FTLR: if $X_1, \ldots, X_n \mid \Gamma \vdash a : A$ is derivable in polymorphic ordered STLC, and M is *any* monoid, then:

The Fundamental Theorem

FTLR: if $X_1, \ldots, X_n \mid \Gamma \vdash a : A$ is derivable in polymorphic ordered STLC, and M is *any* monoid, then:

- For all closed types $B_1, \ldots, B_n$ with relations $R_i \subseteq M \times [\![B_i]\!]$ along with $m \in M$ and $\gamma : \Gamma[B_1/X_1, \ldots, B_n/X_n]$

The Fundamental Theorem

FTLR: if $X_1, \ldots, X_n \mid \Gamma \vdash a : A$ is derivable in polymorphic ordered STLC, and M is *any* monoid, then:

- For all closed types $B_1, \ldots, B_n$ with relations $R_i \subseteq M \times [\![B_i]\!]$ along with $m \in M$ and $\gamma : \Gamma[B_1/X_1, \ldots, B_n/X_n]$

$$\text{if } \; m \Vdash_{\Gamma}^{X_i \mapsto R_i} \gamma \; \text{ then } \; m \Vdash_{A}^{X_i \mapsto R_i} a[\gamma/\Gamma]$$

The Fundamental Theorem

FTLR: if $X_1, \ldots, X_n \mid \Gamma \vdash a : A$ is derivable in polymorphic ordered STLC, and M is *any* monoid, then:

- For all closed types $B_1, \ldots, B_n$ with relations $R_i \subseteq M \times [\![B_i]\!]$ along with $m \in M$ and $\gamma : \Gamma[B_1/X_1, \ldots, B_n/X_n]$

$$\text{if } \; m \Vdash_{\Gamma}^{X_i \mapsto R_i} \gamma \; \text{ then } \; m \Vdash_{A}^{X_i \mapsto R_i} a[\gamma/\Gamma]$$

Proof: Induction on the derivation of $X_1, \ldots, X_n \mid \Gamma \vdash a : A$.

Applications of Substructural Parametricity

Theorem: every closed term $F : \forall X.X/X/(X \otimes X)$ in ordered STLC must be extensionally equivalent to

$$\Lambda X.\lambda x.\lambda y.\langle x, y\rangle$$

Applications of Substructural Parametricity

Theorem: every closed term $F : \forall X.X/X/(X \otimes X)$ in ordered STLC must be extensionally equivalent to

$$\Lambda X.\lambda x.\lambda y.\langle x, y\rangle$$

Proof: let M be the free monoid on two generators α, β, i.e. the set of strings $x_1 x_2 \ldots x_n$ with $x_i \in \{\alpha, \beta\}$, with $\otimes$ given by string concatenation.

Applications of Substructural Parametricity

Theorem: every closed term $F : \forall X.X/X/(X \otimes X)$ in ordered STLC must be extensionally equivalent to

$$\Lambda X.\lambda x.\lambda y.\langle x, y\rangle$$

Proof: let M be the free monoid on two generators α, β, i.e. the set of strings $x_1 x_2 \ldots x_n$ with $x_i \in \{\alpha, \beta\}$, with $\otimes$ given by string concatenation.

Then for any type A Type with $a_0, a_1 : A$, define $R \subseteq M \times [\![A]\!]$ by $m \mathrel{R} a \iff$ either $m = \alpha$ and $a = a_0$ or $m = \beta$ and $a = a_1$.

Applications of Substructural Parametricity

Theorem: every closed term $F : \forall X.X/X/(X \otimes X)$ in ordered STLC must be extensionally equivalent to

$$\Lambda X.\lambda x.\lambda y.\langle x, y\rangle$$

Proof: let M be the free monoid on two generators α, β, i.e. the set of strings $x_1 x_2 \ldots x_n$ with $x_i \in \{\alpha, \beta\}$, with $\otimes$ given by string concatenation.

Then for any type A Type with $a_0, a_1 : A$, define $R \subseteq M \times [\![A]\!]$ by $m \mathrel{R} a \iff$ either $m = \alpha$ and $a = a_0$ or $m = \beta$ and $a = a_1$.

By parametricity for F, we have the following:

Applications of Substructural Parametricity

Theorem: every closed term $F : \forall X.X/X/(X \otimes X)$ in ordered STLC must be extensionally equivalent to

$$\Lambda X.\lambda x.\lambda y.\langle x, y\rangle$$

Proof: let M be the free monoid on two generators α, β, i.e. the set of strings $x_1 x_2 \ldots x_n$ with $x_i \in \{\alpha, \beta\}$, with $\otimes$ given by string concatenation.

Then for any type A Type with $a_0, a_1 : A$, define $R \subseteq M \times [\![A]\!]$ by $m \mathrel{R} a \iff$ either $m = \alpha$ and $a = a_0$ or $m = \beta$ and $a = a_1$.

By parametricity for F, we have the following:

- For all $m, n \in M$, if $m \mathrel{R} a_0$ and $n \mathrel{R} a_1$, there exist $m', n' \in M$ and $a', a'' : A$ such that $m \otimes n = m' \otimes n'$ and $m' \mathrel{R} a'$ and $n' \mathrel{R} a''$ and $F\ A\ a_0\ a_1 \equiv_A \langle a', a''\rangle$.

Applications of Substructural Parametricity

Theorem: every closed term $F : \forall X.X/X/(X \otimes X)$ in ordered STLC must be extensionally equivalent to

$$\Lambda X.\lambda x.\lambda y.\langle x, y\rangle$$

Proof: let M be the free monoid on two generators α, β, i.e. the set of strings $x_1 x_2 \ldots x_n$ with $x_i \in \{\alpha, \beta\}$, with $\otimes$ given by string concatenation.

Then for any type A Type with $a_0, a_1 : A$, define $R \subseteq M \times [\![A]\!]$ by $m \mathrel{R} a \iff$ either $m = \alpha$ and $a = a_0$ or $m = \beta$ and $a = a_1$.

By parametricity for F, we have the following:

- For all $m, n \in M$, if $m \mathrel{R} a_0$ and $n \mathrel{R} a_1$, there exist $m', n' \in M$ and $a', a'' : A$ such that $m \otimes n = m' \otimes n'$ and $m' \mathrel{R} a'$ and $n' \mathrel{R} a''$ and $F\ A\ a_0\ a_1 \equiv_A \langle a', a''\rangle$.
- Substituting α for m and β for n, this implies that we have $m', n' \in \{\alpha, \beta\}$ such that $\alpha\beta = m' \otimes n'$, which implies that $m' = \alpha$ and $n' = \beta$, and therefore $a' = a_0$ and $a'' = a_1$, i.e. $\alpha\ A\ a_0\ a_1 \equiv_A \langle a_0, a_1\rangle$.

How did I do that?

Categories & Languages

Definition: a *category* $\mathcal{C}$ consists of a set of *objects*, and for each pair of objects a set of *arrows* or *morphisms* between them.

$$\mathrm{A}, \mathrm{B} \in \mathcal{C} \qquad f : \mathrm{A} \longrightarrow \mathrm{B} \in \mathcal{C}$$

Categories & Languages

Definition: a *category* $\mathcal{C}$ consists of a set of *objects*, and for each pair of objects a set of *arrows* or *morphisms* between them.

$$A, B \in \mathcal{C} \qquad f : A \longrightarrow B \in \mathcal{C}$$

such that arrows are closed under identity and composition

$$\mathrm{id}_A : A \longrightarrow A \in \mathcal{C} \qquad \begin{array}{l} f : A \longrightarrow B \in \mathcal{C},\ g : B \longrightarrow C \in \mathcal{C} \\ \vdash g \circ f : A \longrightarrow C \in \mathcal{C} \end{array}$$

Categories & Languages

Definition: a *category* $\mathcal{C}$ consists of a set of *objects*, and for each pair of objects a set of *arrows* or *morphisms* between them.

$$\mathrm{A}, \mathrm{B} \in \mathcal{C} \qquad f : \mathrm{A} \longrightarrow \mathrm{B} \in \mathcal{C}$$

such that arrows are closed under identity and composition

$$\mathrm{id}_{\mathrm{A}} : \mathrm{A} \longrightarrow \mathrm{A} \in \mathcal{C} \qquad \begin{array}{l} f : \mathrm{A} \longrightarrow \mathrm{B} \in \mathcal{C},\ g : \mathrm{B} \longrightarrow \mathrm{C} \in \mathcal{C} \\ \vdash g \circ f : \mathrm{A} \longrightarrow \mathrm{C} \in \mathcal{C} \end{array}$$

$$f \circ \mathrm{id}_{\mathrm{A}} = f = \mathrm{id}_{\mathrm{B}} \circ f \qquad (h \circ g) \circ f = h \circ (g \circ f)$$

Categories & Languages

Definition: a *category* $\mathcal{C}$ consists of a set of *objects*, and for each pair of objects a set of *arrows* or *morphisms* between them.

$$\mathrm{A}, \mathrm{B} \in \mathcal{C} \qquad f : \mathrm{A} \longrightarrow \mathrm{B} \in \mathcal{C}$$

such that arrows are closed under identity and composition

$$\mathrm{id}_{\mathrm{A}} : \mathrm{A} \longrightarrow \mathrm{A} \in \mathcal{C} \qquad \begin{array}{l} f : \mathrm{A} \longrightarrow \mathrm{B} \in \mathcal{C},\ g : \mathrm{B} \longrightarrow \mathrm{C} \in \mathcal{C} \\ \vdash g \circ f : \mathrm{A} \longrightarrow \mathrm{C} \in \mathcal{C} \end{array}$$

$$f \circ \mathrm{id}_{\mathrm{A}} = f = \mathrm{id}_{\mathrm{B}} \circ f \qquad (h \circ g) \circ f = h \circ (g \circ f)$$

Example: For any type system $\mathbb{T}$, there is a category $\mathsf{Syn}_{\mathbb{T}}$ – the *syntactic category* of $\mathbb{T}$ – whose objects are types in in $\mathbb{T}$ and whose morphisms $\mathrm{A} \longrightarrow \mathrm{B}$ are terms $x : \mathrm{A} \vdash b : \mathrm{B}$ in $\mathbb{T}$, quotiented up to judgmental equality in $\mathbb{T}$.

Universal Properties

One advantage of the language of category theory is that it lets us straightforwardly characterize various mathematical constructions by how they relate to other objects in the same category:

Universal Properties

One advantage of the language of category theory is that it lets us straightforwardly characterize various mathematical constructions by how they relate to other objects in the same category:

Initial and terminal objects: an object $\bot$ in a category $\mathcal{C}$ is *initial* if for every other object $A \in \mathcal{C}$ there is a unique morphism $*_A$ from $\bot$ to A. Dually, an object $\top$ is *terminal* if for every object $A \in \mathcal{C}$, there is a unique morphism $!_A$ from A to $\top$.

$$*_A : \bot \to A \qquad !_A : A \to \top$$

Universal Properties

One advantage of the language of category theory is that it lets us straightforwardly characterize various mathematical constructions by how they relate to other objects in the same category:

Initial and terminal objects: an object $\bot$ in a category $\mathcal{C}$ is *initial* if for every other object $A \in \mathcal{C}$ there is a unique morphism $*_A$ from $\bot$ to A. Dually, an object $\top$ is *terminal* if for every object $A \in \mathcal{C}$, there is a unique morphism $!_A$ from A to $\top$.

$$*_A : \bot \to A \qquad !_A : A \to \top$$

Example: The unit type **1** is the terminal object in $\mathsf{Syn}_{\mathrm{STLC}}$.

Products

Given objects $A, B \in \mathcal{C}$, the *product* of A and B is an object $A \times B$ with projection morphisms $A \xleftarrow{\pi_1} A \times B \xrightarrow{\pi_2} B$ such that:

Products

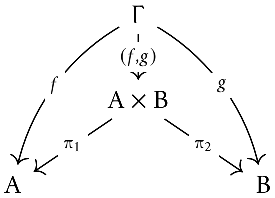
Given objects $A, B \in \mathcal{C}$, the *product* of A and B is an object $A \times B$ with projection morphisms $A \xleftarrow{\pi_1} A \times B \xrightarrow{\pi_2} B$ such that:

- For any other object Γ with morphisms $A \xleftarrow{f} \Gamma \xrightarrow{g} B$, there is a unique morphism $(f, g) : \Gamma \to A \times B$ such that $f = \pi_1 \circ (f, g)$ and $g = \pi_2 \circ (f, g)$. This can be visualized as the following *commutative diagram*.

Products

Given objects $A, B \in \mathcal{C}$, the *product* of A and B is an object $A \times B$ with projection morphisms $A \xleftarrow{\pi_1} A \times B \xrightarrow{\pi_2} B$ such that:

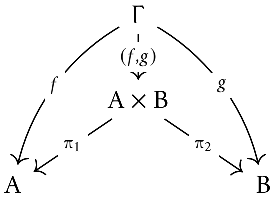
- For any other object Γ with morphisms $A \xleftarrow{f} \Gamma \xrightarrow{g} B$, there is a unique morphism $(f, g) : \Gamma \to A \times B$ such that $f = \pi_1 \circ (f, g)$ and $g = \pi_2 \circ (f, g)$. This can be visualized as the following *commutative diagram*.



Products

Given objects $A, B \in \mathcal{C}$, the *product* of A and B is an object $A \times B$ with projection morphisms $A \xleftarrow{\pi_1} A \times B \xrightarrow{\pi_2} B$ such that:

- For any other object Γ with morphisms $A \xleftarrow{f} \Gamma \xrightarrow{g} B$, there is a unique morphism $(f, g) : \Gamma \to A \times B$ such that $f = \pi_1 \circ (f, g)$ and $g = \pi_2 \circ (f, g)$. This can be visualized as the following *commutative diagram*.



Example: In $\mathsf{Syn}_{\mathrm{STLC}}$, the product of two types A, B is given by the product type $A \times B$.

Exponentials & Natural Numbers Objects

In a category with products $\mathcal{C}$, given two objects $A, B \in \mathcal{C}$, the *exponential* of A and B is an object B^A with a morphism $\alpha : B^A \times A \longrightarrow B$ such that:

Exponentials & Natural Numbers Objects

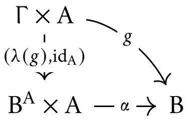
In a category with products $\mathcal{C}$, given two objects $A, B \in \mathcal{C}$, the *exponential* of A and B is an object B^A with a morphism $\alpha : B^A \times A \longrightarrow B$ such that:

- For any other object Γ with $g : \Gamma \times A \longrightarrow B$, there is a unique morphism $\lambda(g) : \Gamma \longrightarrow B^A$ that makes the following diagram commute:

Exponentials & Natural Numbers Objects

In a category with products $\mathcal{C}$, given two objects $A, B \in \mathcal{C}$, the *exponential* of A and B is an object B^A with a morphism $\alpha : B^A \times A \longrightarrow B$ such that:

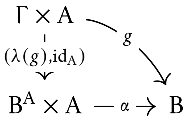
- For any other object Γ with $g : \Gamma \times A \longrightarrow B$, there is a unique morphism $\lambda(g) : \Gamma \longrightarrow B^A$ that makes the following diagram commute:



Exponentials & Natural Numbers Objects

In a category with products $\mathcal{C}$, given two objects $A, B \in \mathcal{C}$, the *exponential* of A and B is an object B^A with a morphism $\alpha : B^A \times A \to B$ such that:

- For any other object Γ with $g : \Gamma \times A \to B$, there is a unique morphism $\lambda(g) : \Gamma \to B^A$ that makes the following diagram commute:

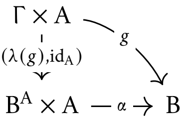


Similarly, if $\mathcal{C}$ additionally has a terminal object $\top$, then a *natural numbers object* (NNO) is an object $\mathbb{N} \in \mathcal{C}$ with $0 : \top \to \mathbb{N}$ and $\mathbf{s} : \mathbb{N} \to \mathbb{N}$ such that:

Exponentials & Natural Numbers Objects

In a category with products $\mathcal{C}$, given two objects $A, B \in \mathcal{C}$, the *exponential* of A and B is an object B^A with a morphism $\alpha : B^A \times A \to B$ such that:

- For any other object Γ with $g : \Gamma \times A \to B$, there is a unique morphism $\lambda(g) : \Gamma \to B^A$ that makes the following diagram commute:



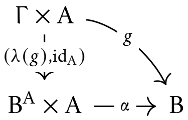
Similarly, if $\mathcal{C}$ additionally has a terminal object $\top$, then a *natural numbers object* (NNO) is an object $\mathbb{N} \in \mathcal{C}$ with $0 : \top \to \mathbb{N}$ and $\mathbf{s} : \mathbb{N} \to \mathbb{N}$ such that:

- For any $A \in \mathcal{C}$ with $a_0 : \top \to A$ and $a_1 : \mathbb{N} \times A \to A$, there is a unique morphism $\mathsf{rec}(a_0, a_1) : \mathbb{N} \to A$ making the following diagrams commute:

Exponentials & Natural Numbers Objects

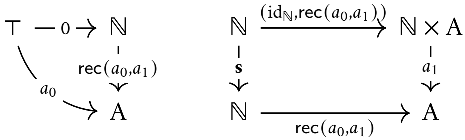
In a category with products $\mathcal{C}$, given two objects $A, B \in \mathcal{C}$, the *exponential* of A and B is an object B^A with a morphism $\alpha : B^A \times A \to B$ such that:

- For any other object Γ with $g : \Gamma \times A \to B$, there is a unique morphism $\lambda(g) : \Gamma \to B^A$ that makes the following diagram commute:



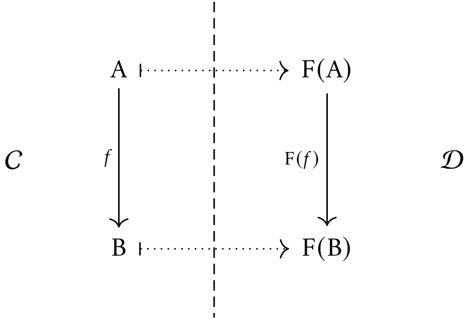
Similarly, if $\mathcal{C}$ additionally has a terminal object $\top$, then a *natural numbers object* (NNO) is an object $\mathbb{N} \in \mathcal{C}$ with $0 : \top \to \mathbb{N}$ and $\mathbf{s} : \mathbb{N} \to \mathbb{N}$ such that:

- For any $A \in \mathcal{C}$ with $a_0 : \top \to A$ and $a_1 : \mathbb{N} \times A \to A$, there is a unique morphism $\mathsf{rec}(a_0, a_1) : \mathbb{N} \to A$ making the following diagrams commute:



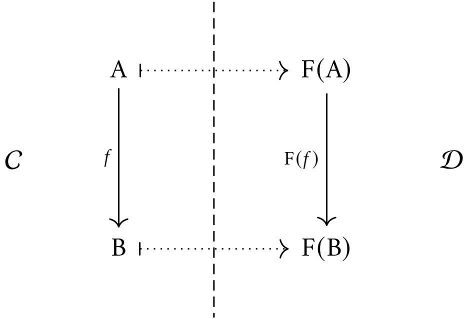
Functors

Given categories $\mathcal{C}, \mathcal{D}$, a *functor* maps objects in $\mathcal{C}$ to objects in $\mathcal{D}$, and morphisms in $\mathcal{C}$ to morphisms in $\mathcal{D}$, as depicted below:



Functors

Given categories $\mathcal{C}, \mathcal{D}$, a *functor* maps objects in $\mathcal{C}$ to objects in $\mathcal{D}$, and morphisms in $\mathcal{C}$ to morphisms in $\mathcal{D}$, as depicted below:

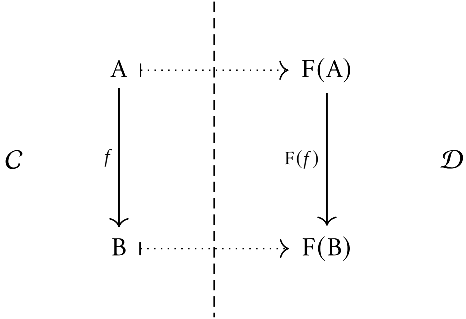


in a manner which preserves identities and composition of morphisms, i.e.

$$\mathrm{F}(\mathrm{id}_{\mathrm{A}}) = \mathrm{id}_{\mathrm{F}(\mathrm{A})} \qquad \mathrm{F}(g \circ f) = \mathrm{F}(g) \circ \mathrm{F}(f)$$

Functors

Given categories $\mathcal{C}, \mathcal{D}$, a *functor* maps objects in $\mathcal{C}$ to objects in $\mathcal{D}$, and morphisms in $\mathcal{C}$ to morphisms in $\mathcal{D}$, as depicted below:



in a manner which preserves identities and composition of morphisms, i.e.

$$F(\mathrm{id}_A) = \mathrm{id}_{F(A)} \qquad F(g \circ f) = F(g) \circ F(f)$$

Note that functors compose associatively and unitally – hence there is a category **Cat** whose objects are categories and whose morphisms are functors.

Initiality of $\mathsf{Syn}_{\mathrm{T}}$

A category $\mathcal{C}$ with a terminal object, products, and exponentials is called *Cartesian Closed*. A functor $\mathrm{F} : \mathcal{C} \to \mathcal{D}$ is called (strictly) Cartesian Closed if it preserves the terminal object, products, and exponentials, i.e.

$$\mathrm{F}(\top) = \top \qquad \mathrm{F}(\mathrm{A} \times \mathrm{B}) = \mathrm{F}(\mathrm{A}) \times \mathrm{F}(\mathrm{B}) \qquad \mathrm{F}(\mathrm{B}^{\mathrm{A}}) = \mathrm{F}(\mathrm{B})^{\mathrm{F}(\mathrm{A})}$$

Initiality of Syn_T

A category $\mathcal{C}$ with a terminal object, products, and exponentials is called *Cartesian Closed*. A functor $F : \mathcal{C} \to \mathcal{D}$ is called (strictly) Cartesian Closed if it preserves the terminal object, products, and exponentials, i.e.

$$F(\top) = \top \qquad F(A \times B) = F(A) \times F(B) \qquad F(B^A) = F(B)^{F(A)}$$

Similarly, if $\mathcal{C}, \mathcal{D}$ have NNOs, then F (strictly) preserves these if $F(\mathbb{N}) = \mathbb{N}$.

Initiality of Syn_T

A category $\mathcal{C}$ with a terminal object, products, and exponentials is called *Cartesian Closed*. A functor $F : \mathcal{C} \to \mathcal{D}$ is called (strictly) Cartesian Closed if it preserves the terminal object, products, and exponentials, i.e.

$$F(\top) = \top \qquad F(A \times B) = F(A) \times F(B) \qquad F(B^A) = F(B)^{F(A)}$$

Similarly, if $\mathcal{C}, \mathcal{D}$ have NNOs, then F (strictly) preserves these if $F(\mathbb{N}) = \mathbb{N}$.

Let $\mathbf{CCC}^{\mathbb{N}}$ be the category with objects Cartesian Closed Categories with NNOs, and morphisms Cartesian Closed functors that preserve NNOs.

Initiality of Syn_T

A category $\mathcal{C}$ with a terminal object, products, and exponentials is called *Cartesian Closed*. A functor $F : \mathcal{C} \to \mathcal{D}$ is called (strictly) Cartesian Closed if it preserves the terminal object, products, and exponentials, i.e.

$$F(\top) = \top \qquad F(A \times B) = F(A) \times F(B) \qquad F(B^A) = F(B)^{F(A)}$$

Similarly, if $\mathcal{C}, \mathcal{D}$ have NNOs, then F (strictly) preserves these if $F(\mathbb{N}) = \mathbb{N}$.

Let $\mathbf{CCC}^{\mathbb{N}}$ be the category with objects Cartesian Closed Categories with NNOs, and morphisms Cartesian Closed functors that preserve NNOs.

Theorem: Syn_T, the syntactic category of System T, is initial in $\mathbf{CCC}^{\mathbb{N}}$.

Initiality of Syn_T

A category $\mathcal{C}$ with a terminal object, products, and exponentials is called *Cartesian Closed*. A functor $F : \mathcal{C} \to \mathcal{D}$ is called (strictly) Cartesian Closed if it preserves the terminal object, products, and exponentials, i.e.

$$F(\top) = \top \qquad F(A \times B) = F(A) \times F(B) \qquad F(B^A) = F(B)^{F(A)}$$

Similarly, if $\mathcal{C}, \mathcal{D}$ have NNOs, then F (strictly) preserves these if $F(\mathbb{N}) = \mathbb{N}$.

Let $\mathbf{CCC}^{\mathbb{N}}$ be the category with objects Cartesian Closed Categories with NNOs, and morphisms Cartesian Closed functors that preserve NNOs.

Theorem: Syn_T, the syntactic category of System T, is initial in $\mathbf{CCC}^{\mathbb{N}}$.

Proof: induction on derivations OR sledgehammer with general facts about GATs, etc.

Initiality of $\mathsf{Syn}_{\mathrm{T}}$

A category $\mathcal{C}$ with a terminal object, products, and exponentials is called *Cartesian Closed*. A functor $\mathrm{F} : \mathcal{C} \to \mathcal{D}$ is called (strictly) Cartesian Closed if it preserves the terminal object, products, and exponentials, i.e.

$$\mathrm{F}(\top) = \top \qquad \mathrm{F}(\mathrm{A} \times \mathrm{B}) = \mathrm{F}(\mathrm{A}) \times \mathrm{F}(\mathrm{B}) \qquad \mathrm{F}(\mathrm{B}^{\mathrm{A}}) = \mathrm{F}(\mathrm{B})^{\mathrm{F}(\mathrm{A})}$$

Similarly, if $\mathcal{C}, \mathcal{D}$ have NNOs, then F (strictly) preserves these if $\mathrm{F}(\mathbb{N}) = \mathbb{N}$.

Let $\mathbf{CCC}^{\mathbb{N}}$ be the category with objects Cartesian Closed Categories with NNOs, and morphisms Cartesian Closed functors that preserve NNOs.

Theorem: $\mathsf{Syn}_{\mathrm{T}}$, the syntactic category of System T, is initial in $\mathbf{CCC}^{\mathbb{N}}$.

Proof: induction on derivations OR sledgehammer with general facts about GATs, etc.

Idea: reduce FTLR for System T to the existence of a functor $\mathsf{Syn}_{\mathrm{T}} \to \mathsf{Pred}_{\mathrm{T}} \in \mathbf{CCC}^{\mathbb{N}}$, for a suitably constructed category of “logical predicates” $\mathsf{Pred}_{\mathrm{T}}$.

From Logical to Categorical Predicates

Define the category $\mathsf{Pred}_{\mathrm{T}}$ as follows:

From Logical to Categorical Predicates

Define the category $\mathsf{Pred}_{\mathrm{T}}$ as follows:

- Objects are pairs (A, P_A) where A is a type in System T and $P \subseteq [\![A]\!]$.

From Logical to Categorical Predicates

Define the category Pred_T as follows:

- Objects are pairs (A, P_A) where A is a type in System T and $P \subseteq [\![A]\!]$.
- A morphism $(A, P_A) \to (B, P_B)$ is a term $x : A \vdash f : B$ in System T such that for all $a : A$, if $P_A(a)$ then $P_B(f[a/x])$.

From Logical to Categorical Predicates

Define the category Pred_T as follows:

- Objects are pairs (A, P_A) where A is a type in System T and $P \subseteq [\![A]\!]$.
- A morphism $(A, P_A) \to (B, P_B)$ is a term $x : A \vdash f : B$ in System T such that for all $a : A$, if $P_A(a)$ then $P_B(f[a/x])$.

Pred_T is Cartesian Closed and has a natural numbers object as follows:

From Logical to Categorical Predicates

Define the category $\mathsf{Pred}_{\mathrm{T}}$ as follows:

- Objects are pairs (A, P_A) where A is a type in System T and $P \subseteq [\![A]\!]$.
- A morphism $(A, P_A) \to (B, P_B)$ is a term $x : A \vdash f : B$ in System T such that for all $a : A$, if $P_A(a)$ then $P_B(f[a/x])$.

$\mathsf{Pred}_{\mathrm{T}}$ is Cartesian Closed and has a natural numbers object as follows:

- The terminal object of $\mathsf{Pred}_{\mathrm{T}}$ is the pair $(\mathbf{1}, \{()\})$

From Logical to Categorical Predicates

Define the category Pred_{T} as follows:

- Objects are pairs (A, P_A) where A is a type in System T and $P \subseteq [\![A]\!]$.
- A morphism $(A, P_A) \to (B, P_B)$ is a term $x : A \vdash f : B$ in System T such that for all $a : A$, if $P_A(a)$ then $P_B(f[a/x])$.

Pred_{T} is Cartesian Closed and has a natural numbers object as follows:

- The terminal object of Pred_{T} is the pair $(\mathbf{1}, \{()\})$
- The product of (A, P_A) and (B, P_B) is given by $(A \times B, P_{A\times B})$, where

$$P_{A\times B}(p) \iff P_A(\pi_1(p)) \text{ and } P_B(\pi_2(p))$$

From Logical to Categorical Predicates

Define the category Pred_T as follows:

- Objects are pairs (A, P_A) where A is a type in System T and $P \subseteq [\![A]\!]$.
- A morphism $(A, P_A) \to (B, P_B)$ is a term $x : A \vdash f : B$ in System T such that for all $a : A$, if $P_A(a)$ then $P_B(f[a/x])$.

Pred_T is Cartesian Closed and has a natural numbers object as follows:

- The terminal object of Pred_T is the pair $(\mathbf{1}, \{()\})$
- The product of (A, P_A) and (B, P_B) is given by $(A \times B, P_{A\times B})$, where

$$P_{A\times B}(p) \iff P_A(\pi_1(p)) \text{ and } P_B(\pi_2(p))$$

- The exponential of (A, P_A) and (B, P_B) is given by $(A \to B, P_{A\to B})$, where

$$P_{A\to B}(f) \iff \forall a : A.P_A(a) \implies P_B(f(a))$$

From Logical to Categorical Predicates

Define the category Pred_T as follows:

- Objects are pairs (A, P_A) where A is a type in System T and $P \subseteq [\![A]\!]$.
- A morphism $(A, P_A) \to (B, P_B)$ is a term $x : A \vdash f : B$ in System T such that for all $a : A$, if $P_A(a)$ then $P_B(f[a/x])$.

Pred_T is Cartesian Closed and has a natural numbers object as follows:

- The terminal object of Pred_T is the pair $(\mathbf{1}, \{()\})$
- The product of (A, P_A) and (B, P_B) is given by $(A \times B, P_{A\times B})$, where

$$P_{A\times B}(p) \iff P_A(\pi_1(p)) \text{ and } P_B(\pi_2(p))$$

- The exponential of (A, P_A) and (B, P_B) is given by $(A \to B, P_{A\to B})$, where

$$P_{A\to B}(f) \iff \forall a : A.P_A(a) \implies P_B(f(a))$$

- The natural numbers object of Pred_T is given by $(\mathbb{N}, P_{\mathbb{N}})$ where

$$P_{\mathbb{N}}(n) \iff \exists m \text{ s.t. } n \equiv_{\mathbb{N}} \overline{m}$$

FLTR, Categorically

By construction, there is a functor $\pi : \mathsf{Pred}_T \to \mathsf{Syn}_T \in \mathbf{CCC}^{\mathbb{N}}$ such that

$$\pi(A, P_A) = A$$

FLTR, Categorically

By construction, there is a functor $\pi : \mathsf{Pred}_T \to \mathsf{Syn}_T \in \mathbf{CCC}^{\mathbb{N}}$ such that

$$\pi(A, P_A) = A$$

Then since $\mathsf{Pred}_T \in \mathbb{CCC}^{\mathbb{N}}$, it follows that there is also a functor $\mathbb{P} : \mathsf{Syn}_T \to \mathsf{Pred}_T \in \mathbf{CCC}^{\mathbb{N}}$.

FLTR, Categorically

By construction, there is a functor $\pi : \mathsf{Pred}_T \to \mathsf{Syn}_T \in \mathbf{CCC}^{\mathbb{N}}$ such that

$$\pi(A, P_A) = A$$

Then since $\mathsf{Pred}_T \in \mathbb{CCC}^{\mathbb{N}}$, it follows that there is also a functor $\mathbb{P} : \mathsf{Syn}_T \to \mathsf{Pred}_T \in \mathbf{CCC}^{\mathbb{N}}$.

Hence we have $\pi \circ \mathbb{P} : \mathsf{Syn}_T \to \mathsf{Syn}_T \in \mathbf{CCC}^{\mathbb{N}}$.

FLTR, Categorically

By construction, there is a functor $\pi : \mathsf{Pred}_T \to \mathsf{Syn}_T \in \mathbf{CCC}^{\mathbb{N}}$ such that

$$\pi(A, P_A) = A$$

Then since $\mathsf{Pred}_T \in \mathbb{CCC}^{\mathbb{N}}$, it follows that there is also a functor $\mathbb{P} : \mathsf{Syn}_T \to \mathsf{Pred}_T \in \mathbf{CCC}^{\mathbb{N}}$.

Hence we have $\pi \circ \mathbb{P} : \mathsf{Syn}_T \to \mathsf{Syn}_T \in \mathbf{CCC}^{\mathbb{N}}$.

But since Syn_T is initial in $\mathbf{CCC}^{\mathbb{N}}$, there can only be one such functor $\mathsf{Syn}_T \to \mathsf{Syn}_T$, so we must have $\pi \circ \mathbb{P} = \mathrm{Id}_{\mathsf{Syn}_T}$, i.e.

$$\begin{array}{ccc} \mathsf{Syn}_T & \xrightarrow{\mathbb{P}} & \mathsf{Pred}_T \\ & \diagdown\!\!\diagdown & \downarrow \pi \\ & & \mathsf{Syn}_T \end{array}$$

FLTR, Categorically

By construction, there is a functor $\pi : \mathsf{Pred}_T \to \mathsf{Syn}_T \in \mathbf{CCC}^{\mathbb{N}}$ such that

$$\pi(A, P_A) = A$$

Then since $\mathsf{Pred}_T \in \mathbb{CCC}^{\mathbb{N}}$, it follows that there is also a functor $\mathbb{P} : \mathsf{Syn}_T \to \mathsf{Pred}_T \in \mathbf{CCC}^{\mathbb{N}}$.

Hence we have $\pi \circ \mathbb{P} : \mathsf{Syn}_T \to \mathsf{Syn}_T \in \mathbf{CCC}^{\mathbb{N}}$.

But since Syn_T is initial in $\mathbf{CCC}^{\mathbb{N}}$, there can only be one such functor $\mathsf{Syn}_T \to \mathsf{Syn}_T$, so we must have $\pi \circ \mathbb{P} = \mathrm{Id}_{\mathsf{Syn}_T}$, i.e.

$$\begin{array}{ccc} \mathsf{Syn}_T & \xrightarrow{\mathbb{P}} & \mathsf{Pred}_T \\ & \diagdown\!\!\diagdown & \big\downarrow \pi \\ & & \mathsf{Syn}_T \end{array}$$

Claim: the fact that the above diagram commutes is equivalent to FTLR.

FLTR, Categorically

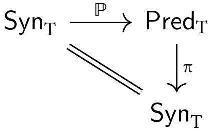
By construction, there is a functor $\pi : \mathsf{Pred}_T \to \mathsf{Syn}_T \in \mathbf{CCC}^{\mathbb{N}}$ such that

$$\pi(A, P_A) = A$$

Then since $\mathsf{Pred}_T \in \mathbb{CCC}^{\mathbb{N}}$, it follows that there is also a functor $\mathbb{P} : \mathsf{Syn}_T \to \mathsf{Pred}_T \in \mathbf{CCC}^{\mathbb{N}}$.

Hence we have $\pi \circ \mathbb{P} : \mathsf{Syn}_T \to \mathsf{Syn}_T \in \mathbf{CCC}^{\mathbb{N}}$.

But since Syn_T is initial in $\mathbf{CCC}^{\mathbb{N}}$, there can only be one such functor $\mathsf{Syn}_T \to \mathsf{Syn}_T$, so we must have $\pi \circ \mathbb{P} = \mathrm{Id}_{\mathsf{Syn}_T}$, i.e.



Claim: the fact that the above diagram commutes is equivalent to FTLR.

- By the above, $\mathbb{P}$ maps each term $x : A \vdash f : B$ in System T to a morphism $f' : (A, \mathbb{P}_A) \to (B, \mathbb{P}_B) \in \mathsf{Pred}_T$ such that $\pi(f') = f$.

FLTR, Categorically

By construction, there is a functor $\pi : \mathsf{Pred}_T \to \mathsf{Syn}_T \in \mathbf{CCC}^{\mathbb{N}}$ such that

$$\pi(A, P_A) = A$$

Then since $\mathsf{Pred}_T \in \mathbb{CCC}^{\mathbb{N}}$, it follows that there is also a functor $\mathbb{P} : \mathsf{Syn}_T \to \mathsf{Pred}_T \in \mathbf{CCC}^{\mathbb{N}}$.

Hence we have $\pi \circ \mathbb{P} : \mathsf{Syn}_T \to \mathsf{Syn}_T \in \mathbf{CCC}^{\mathbb{N}}$.

But since Syn_T is initial in $\mathbf{CCC}^{\mathbb{N}}$, there can only be one such functor $\mathsf{Syn}_T \to \mathsf{Syn}_T$, so we must have $\pi \circ \mathbb{P} = \mathrm{Id}_{\mathsf{Syn}_T}$, i.e.

$$\begin{array}{ccc} \mathsf{Syn}_T & \xrightarrow{\mathbb{P}} & \mathsf{Pred}_T \\ & \diagdown\!\!\diagdown & \big\downarrow \pi \\ & & \mathsf{Syn}_T \end{array}$$

Claim: the fact that the above diagram commutes is equivalent to FTLR.

- By the above, $\mathbb{P}$ maps each term $x : A \vdash f : B$ in System T to a morphism $f' : (A, \mathbb{P}_A) \to (B, \mathbb{P}_B) \in \mathsf{Pred}_T$ such that $\pi(f') = f$.
  - i.e. for all $a : A$, if $\mathbb{P}_A(a)$ then $\mathbb{P}_B(f[a/x])$.

A Recipe for Logical Relations

To define logical relations and prove FTLR for a given type system $\mathbb{T}$

A Recipe for Logical Relations

To define logical relations and prove FTLR for a given type system $\mathbb{T}$

1. Show that $\mathsf{Syn}_{\mathbb{T}}$ is the initial object in some category of structured categories $\mathcal{C}$.

A Recipe for Logical Relations

To define logical relations and prove FTLR for a given type system $\mathbb{T}$

1. Show that $\mathsf{Syn}_{\mathbb{T}}$ is the initial object in some category of structured categories $\mathcal{C}$.
    - If you play your cards right, this is the easy part.

A Recipe for Logical Relations

To define logical relations and prove FTLR for a given type system $\mathbb{T}$

1. Show that $\mathsf{Syn}_{\mathbb{T}}$ is the initial object in some category of structured categories $\mathcal{C}$.
    - If you play your cards right, this is the easy part.
2. Build a *category of relations* $\mathsf{Rel}_{\mathbb{T}}$ and show that it is also an object of $\mathcal{C}$ with a morphism $\pi : \mathsf{Rel}_{\mathbb{T}} \to \mathsf{Syn}_{\mathbb{T}}$.

A Recipe for Logical Relations

To define logical relations and prove FTLR for a given type system $\mathbb{T}$

1. Show that $\mathsf{Syn}_{\mathbb{T}}$ is the initial object in some category of structured categories $\mathcal{C}$.
   - If you play your cards right, this is the easy part.
2. Build a *category of relations* $\mathsf{Rel}_{\mathbb{T}}$ and show that it is also an object of $\mathcal{C}$ with a morphism $\pi : \mathsf{Rel}_{\mathbb{T}} \to \mathsf{Syn}_{\mathbb{T}}$.
3. Derive FTLR from the existence of a commuting triangle of functors as below:

$$\begin{array}{ccc} \mathsf{Syn}_{\mathbb{T}} & \longrightarrow & \mathsf{Rel}_{\mathbb{T}} \\ & \diagdown\!\!\diagdown & \big\downarrow \pi \\ & & \mathsf{Syn}_{\mathbb{T}} \end{array}$$

Monoidal Categories & Ordered STLC

A *monoidal category* is a category $\mathcal{M}$ equipped with an object $\epsilon \in \mathcal{M}$ and a functor $\otimes : \mathcal{M} \times \mathcal{M} \to \mathcal{M}$ with natural isomorphisms

$$\epsilon \otimes A \cong A \cong A \otimes \epsilon \qquad (A \otimes B) \otimes C \cong A \otimes (B \otimes C)$$

subject to certain coherence conditions.

Monoidal Categories & Ordered STLC

A *monoidal category* is a category $\mathcal{M}$ equipped with an object $\epsilon \in \mathcal{M}$ and a functor $\otimes : \mathcal{M} \times \mathcal{M} \to \mathcal{M}$ with natural isomorphisms

$$\epsilon \otimes A \cong A \cong A \otimes \epsilon \qquad (A \otimes B) \otimes C \cong A \otimes (B \otimes C)$$

subject to certain coherence conditions.

We say that $\mathcal{M}$ is *biclosed* if for all $A, B \in \mathcal{M}$ there are objects A/B and $B \setminus A$ with morphisms $\alpha_L : A \otimes A/B \to B$ and $\alpha_R : B \setminus A \otimes A \to B$ such that:

Monoidal Categories & Ordered STLC

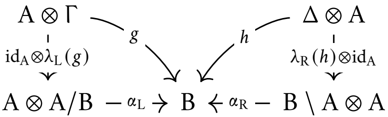
A *monoidal category* is a category $\mathcal{M}$ equipped with an object $\epsilon \in \mathcal{M}$ and a functor $\otimes : \mathcal{M} \times \mathcal{M} \to \mathcal{M}$ with natural isomorphisms

$$\epsilon \otimes A \cong A \cong A \otimes \epsilon \qquad (A \otimes B) \otimes C \cong A \otimes (B \otimes C)$$

subject to certain coherence conditions.

We say that $\mathcal{M}$ is *biclosed* if for all $A, B \in \mathcal{M}$ there are objects A/B and $B \setminus A$ with morphisms $\alpha_L : A \otimes A/B \to B$ and $\alpha_R : B \setminus A \otimes A \to B$ such that:

- For all $\Gamma, \Delta \in \mathcal{M}$ with $g : A \otimes \Gamma \to B$ and $h : \Delta \otimes A \to B$, there are morphisms $\lambda_L(g) : \Gamma \to A/B$ and $\lambda_R(h) : \Delta \to B \setminus A$ making the following commute:



Monoidal Categories & Ordered STLC

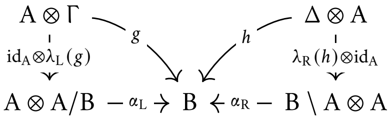
A *monoidal category* is a category $\mathcal{M}$ equipped with an object $\epsilon \in \mathcal{M}$ and a functor $\otimes : \mathcal{M} \times \mathcal{M} \to \mathcal{M}$ with natural isomorphisms

$$\epsilon \otimes A \cong A \cong A \otimes \epsilon \qquad (A \otimes B) \otimes C \cong A \otimes (B \otimes C)$$

subject to certain coherence conditions.

We say that $\mathcal{M}$ is *biclosed* if for all $A, B \in \mathcal{M}$ there are objects A/B and $B \setminus A$ with morphisms $\alpha_L : A \otimes A/B \to B$ and $\alpha_R : B \setminus A \otimes A \to B$ such that:

- For all $\Gamma, \Delta \in \mathcal{M}$ with $g : A \otimes \Gamma \to B$ and $h : \Delta \otimes A \to B$, there are morphisms $\lambda_L(g) : \Gamma \to A/B$ and $\lambda_R(h) : \Delta \to B \setminus A$ making the following commute:



Theorem: The syntactic category of ordered STLC is initial in the category of biclosed monoidal categories and functors that preserve ϵ, $\otimes$, $/$, and $\setminus$.

Substructural Categorical Relations

Fix a monoid M. Define a category $\mathsf{Rel}_{\otimes}^{M}$ as follows:

Substructural Categorical Relations

Fix a monoid M. Define a category $\mathsf{Rel}_{\otimes}^{M}$ as follows:

- Objects are pairs $(A, \Vdash_A)$ where A is a type in ordered STLC and $\Vdash_A \subseteq M \times [\![A]\!]$

Substructural Categorical Relations

Fix a monoid M. Define a category $\mathsf{Rel}_{\otimes}^{M}$ as follows:

- Objects are pairs $(A, \Vdash_A)$ where A is a type in ordered STLC and $\Vdash_A \subseteq M \times [\![A]\!]$
- A morphism $(A, \Vdash_A) \longrightarrow (B, \Vdash_B)$ is a term $x : A \vdash f : B$ in ordered STLC such that for all $m \in M$ and $a : A$, if $m \Vdash_A a$ then $m \Vdash_B f[a/x]$

Substructural Categorical Relations

Fix a monoid M. Define a category $\mathsf{Rel}_{\otimes}^{M}$ as follows:

- Objects are pairs $(A, \Vdash_A)$ where A is a type in ordered STLC and $\Vdash_A \subseteq M \times [\![A]\!]$
- A morphism $(A, \Vdash_A) \to (B, \Vdash_B)$ is a term $x : A \vdash f : B$ in ordered STLC such that for all $m \in M$ and $a : A$, if $m \Vdash_A a$ then $m \Vdash_B f[a/x]$

$\mathsf{Rel}_{\otimes}^{M}$ can be given the structure of a biclosed monoidal category as follows:

Substructural Categorical Relations

Fix a monoid M. Define a category $\mathsf{Rel}_{\otimes}^{\mathrm{M}}$ as follows:

- Objects are pairs $(\mathrm{A}, \Vdash_{\mathrm{A}})$ where A is a type in ordered STLC and $\Vdash_{\mathrm{A}} \subseteq \mathrm{M} \times [\![\mathrm{A}]\!]$
- A morphism $(\mathrm{A}, \Vdash_{\mathrm{A}}) \longrightarrow (\mathrm{B}, \Vdash_{\mathrm{B}})$ is a term $x : \mathrm{A} \vdash f : \mathrm{B}$ in ordered STLC such that for all $m \in \mathrm{M}$ and $a : \mathrm{A}$, if $m \Vdash_{\mathrm{A}} a$ then $m \Vdash_{\mathrm{B}} f[a/x]$

$\mathsf{Rel}_{\otimes}^{\mathrm{M}}$ can be given the structure of a biclosed monoidal category as follows:

- The monoidal unit is $(\mathbf{1}, \{(\epsilon, \langle\rangle)\})$

Substructural Categorical Relations

Fix a monoid M. Define a category $\mathsf{Rel}_{\otimes}^{\mathrm{M}}$ as follows:

- Objects are pairs $(\mathrm{A}, \Vdash_{\mathrm{A}})$ where A is a type in ordered STLC and $\Vdash_{\mathrm{A}} \subseteq \mathrm{M} \times [\![\mathrm{A}]\!]$
- A morphism $(\mathrm{A}, \Vdash_{\mathrm{A}}) \longrightarrow (\mathrm{B}, \Vdash_{\mathrm{B}})$ is a term $x : \mathrm{A} \vdash f : \mathrm{B}$ in ordered STLC such that for all $m \in \mathrm{M}$ and $a : \mathrm{A}$, if $m \Vdash_{\mathrm{A}} a$ then $m \Vdash_{\mathrm{B}} f[a/x]$

$\mathsf{Rel}_{\otimes}^{\mathrm{M}}$ can be given the structure of a biclosed monoidal category as follows:

- The monoidal unit is $(\mathbf{1}, \{(\epsilon, \langle\rangle)\})$
- Given $(\mathrm{A}, \Vdash_{\mathrm{A}})$ and $(\mathrm{B}, \Vdash_{\mathrm{B}})$, define their monoidal product as $(\mathrm{A} \otimes \mathrm{B}, \Vdash_{\mathrm{A}\otimes\mathrm{B}})$ where

$$m \Vdash_{\mathrm{A}\otimes\mathrm{B}} p \iff \begin{array}{l} \exists n, k \in \mathrm{M},\ a : \mathrm{A},\ b : \mathrm{B} \text{ s.t.} \\ m = n \otimes k,\ p \equiv_{\mathrm{A}\otimes\mathrm{B}} \langle a, b\rangle,\ n \Vdash_{\mathrm{A}} a \text{ and } k \Vdash_{\mathrm{B}} b \end{array}$$

Substructural Categorical Relations

Fix a monoid M. Define a category $\mathsf{Rel}^{\mathrm{M}}_{\otimes}$ as follows:

- Objects are pairs $(\mathrm{A}, \Vdash_{\mathrm{A}})$ where A is a type in ordered STLC and $\Vdash_{\mathrm{A}} \subseteq \mathrm{M} \times [\![\mathrm{A}]\!]$
- A morphism $(\mathrm{A}, \Vdash_{\mathrm{A}}) \to (\mathrm{B}, \Vdash_{\mathrm{B}})$ is a term $x : \mathrm{A} \vdash f : \mathrm{B}$ in ordered STLC such that for all $m \in \mathrm{M}$ and $a : \mathrm{A}$, if $m \Vdash_{\mathrm{A}} a$ then $m \Vdash_{\mathrm{B}} f[a/x]$

$\mathsf{Rel}^{\mathrm{M}}_{\otimes}$ can be given the structure of a biclosed monoidal category as follows:

- The monoidal unit is $(\mathbf{1}, \{(\epsilon, \langle\rangle)\})$
- Given $(\mathrm{A}, \Vdash_{\mathrm{A}})$ and $(\mathrm{B}, \Vdash_{\mathrm{B}})$, define their monoidal product as $(\mathrm{A} \otimes \mathrm{B}, \Vdash_{\mathrm{A}\otimes\mathrm{B}})$ where

$$m \Vdash_{\mathrm{A}\otimes\mathrm{B}} p \iff \begin{array}{l} \exists n, k \in \mathrm{M},\ a : \mathrm{A},\ b : \mathrm{B} \text{ s.t.} \\ m = n \otimes k,\ p \equiv_{\mathrm{A}\otimes\mathrm{B}} \langle a, b\rangle,\ n \Vdash_{\mathrm{A}} a \text{ and } k \Vdash_{\mathrm{B}} b \end{array}$$

- Given $(\mathrm{A}, \Vdash_{\mathrm{A}})$ and $(\mathrm{B}, \Vdash_{\mathrm{B}})$, define their left closure as $(\mathrm{A}/\mathrm{B}, \Vdash_{\mathrm{A}/\mathrm{B}})$ where

$$m \Vdash_{\mathrm{A}/\mathrm{B}} f \iff \forall n \in \mathrm{M},\ a : \mathrm{A},\ n \Vdash_{\mathrm{A}} a \implies n \otimes m \Vdash_{\mathrm{B}} f(a)$$

Substructural Categorical Relations

Fix a monoid M. Define a category $\mathsf{Rel}_{\otimes}^{M}$ as follows:

- Objects are pairs $(A, \Vdash_A)$ where A is a type in ordered STLC and $\Vdash_A \subseteq M \times [\![A]\!]$
- A morphism $(A, \Vdash_A) \to (B, \Vdash_B)$ is a term $x : A \vdash f : B$ in ordered STLC such that for all $m \in M$ and $a : A$, if $m \Vdash_A a$ then $m \Vdash_B f[a/x]$

$\mathsf{Rel}_{\otimes}^{M}$ can be given the structure of a biclosed monoidal category as follows:

- The monoidal unit is $(\mathbf{1}, \{(\epsilon, \langle\rangle)\})$
- Given $(A, \Vdash_A)$ and $(B, \Vdash_B)$, define their monoidal product as $(A \otimes B, \Vdash_{A \otimes B})$ where

$$m \Vdash_{A\otimes B} p \iff \begin{array}{l} \exists n, k \in M,\ a : A,\ b : B \text{ s.t.} \\ m = n \otimes k,\ p \equiv_{A\otimes B} \langle a, b\rangle,\ n \Vdash_A a \text{ and } k \Vdash_B b \end{array}$$

- Given $(A, \Vdash_A)$ and $(B, \Vdash_B)$, define their left closure as $(A/B, \Vdash_{A/B})$ where

$$m \Vdash_{A/B} f \iff \forall n \in M,\ a : A,\ n \Vdash_A a \implies n \otimes m \Vdash_B f(a)$$

- And similarly for the right closure.

Adding Polymorphism

The syntactic category of System F is not really one category, but rather a family of Cartesian Closed categories $\mathsf{Syn}[X_1, \ldots, X_n]$ for each list of type variables $X_1, \ldots, X_n$. These come equipped with functors

$$\forall_{X_{n+1}} : \mathsf{Syn}[X_1, \ldots, X_n, X_{n+1}] \to \mathsf{Syn}[X_1, \ldots, X_n]$$

with a certain universal property. Moreover, this family of categories is *initial* with this property.

Adding Polymorphism

The syntactic category of System F is not really one category, but rather a family of Cartesian Closed categories $\mathsf{Syn}[X_1, \ldots, X_n]$ for each list of type variables $X_1, \ldots, X_n$. These come equipped with functors

$$\forall_{X_{n+1}} : \mathsf{Syn}[X_1, \ldots, X_n, X_{n+1}] \to \mathsf{Syn}[X_1, \ldots, X_n]$$

with a certain universal property. Moreover, this family of categories is *initial* with this property.

Idea: for each $X_1, \ldots, X_n$, build a Cartesian Closed category $\mathsf{Rel}[X_1, \ldots, X_n]$ following the same general procedure as for STLC/System T/etc., and then show that there are functors $\forall_{X_{n+1}} : \mathsf{Rel}[X_1, \ldots, X_n, X_{n+1}] \to \mathsf{Rel}[X_1, \ldots, X_n]$ with the requisite universal property, using the definition of logical relations for parametrically polymorphic functions.

Adding Polymorphism

The syntactic category of System F is not really one category, but rather a family of Cartesian Closed categories $\mathsf{Syn}[X_1, \ldots, X_n]$ for each list of type variables $X_1, \ldots, X_n$. These come equipped with functors

$$\forall_{X_{n+1}} : \mathsf{Syn}[X_1, \ldots, X_n, X_{n+1}] \to \mathsf{Syn}[X_1, \ldots, X_n]$$

with a certain universal property. Moreover, this family of categories is *initial* with this property.

Idea: for each $X_1, \ldots, X_n$, build a Cartesian Closed category $\mathsf{Rel}[X_1, \ldots, X_n]$ following the same general procedure as for STLC/System T/etc., and then show that there are functors $\forall_{X_{n+1}} : \mathsf{Rel}[X_1, \ldots, X_n, X_{n+1}] \to \mathsf{Rel}[X_1, \ldots, X_n]$ with the requisite universal property, using the definition of logical relations for parametrically polymorphic functions.

This procedure works just as well when we replace Cartesian Closed categories with biclosed monoidal categories, and thus allows us to generalize parametricity from ordinary System F to substructural variants thereof (and even further beyond!)

Conclusion

Category theory provides a general and powerfully simplifying language for logical relations. In this talk, we have seen how this method can be used to handle logical relations and parametricity for ordered polymorphic λ-calculus, but the applications of this method are seemingly without limit.

Conclusion

Category theory provides a general and powerfully simplifying language for logical relations. In this talk, we have seen how this method can be used to handle logical relations and parametricity for ordered polymorphic λ-calculus, but the applications of this method are seemingly without limit.

- By augmenting the structures of the monoidal categories and monoids used in the above construction for ordered parametricity, we can prove parametricity theorems for other substructrual polymorphic λ-calculi (e.g. using *symmetric monoidal categories* and *commutative monoids* gives parametricity for Linear System F).

Conclusion

Category theory provides a general and powerfully simplifying language for logical relations. In this talk, we have seen how this method can be used to handle logical relations and parametricity for ordered polymorphic λ-calculus, but the applications of this method are seemingly without limit.

- By augmenting the structures of the monoidal categories and monoids used in the above construction for ordered parametricity, we can prove parametricity theorems for other substructrual polymorphic λ-calculi (e.g. using *symmetric monoidal categories* and *commutative monoids* gives parametricity for Linear System F).
- The categorical concepts of *adjoint functors* and *monoidal adjunctions* can be used to extend logical relations and parametricity to systems of *adjoint modal substructural type theory* that allow us to combine different sorts of substructural constraints in the same type system.

Conclusion

Category theory provides a general and powerfully simplifying language for logical relations. In this talk, we have seen how this method can be used to handle logical relations and parametricity for ordered polymorphic λ-calculus, but the applications of this method are seemingly without limit.

- By augmenting the structures of the monoidal categories and monoids used in the above construction for ordered parametricity, we can prove parametricity theorems for other substructrual polymorphic λ-calculi (e.g. using *symmetric monoidal categories* and *commutative monoids* gives parametricity for Linear System F).
- The categorical concepts of *adjoint functors* and *monoidal adjunctions* can be used to extend logical relations and parametricity to systems of *adjoint modal substructural type theory* that allow us to combine different sorts of substructural constraints in the same type system.
- The categorical semantics of *dependent type theory* can be used to extend logical relations and parametricity for systems with dependent types.

Conclusion

Category theory provides a general and powerfully simplifying language for logical relations. In this talk, we have seen how this method can be used to handle logical relations and parametricity for ordered polymorphic λ-calculus, but the applications of this method are seemingly without limit.

- By augmenting the structures of the monoidal categories and monoids used in the above construction for ordered parametricity, we can prove parametricity theorems for other substructural polymorphic λ-calculi (e.g. using *symmetric monoidal categories* and *commutative monoids* gives parametricity for Linear System F).
- The categorical concepts of *adjoint functors* and *monoidal adjunctions* can be used to extend logical relations and parametricity to systems of *adjoint modal substructural type theory* that allow us to combine different sorts of substructural constraints in the same type system.
- The categorical semantics of *dependent type theory* can be used to extend logical relations and parametricity for systems with dependent types.

Food for thought: what makes a category behave like a “category of relations”?

Thank you!