1 Object Language

1.1 Raw Terms

The untyped syntax for CBPV OSum, add separating connectives, remove error.

$$
\begin{array}{llcl}
\text{Value Types} & A & ::= & X \\
 & & \mid & \text{Unit} \\
 & & \mid & \text{Case } A \\
 & & \mid & \text{OSum} \\
 & & \mid & A \times A \\
 & & \mid & A * A \\
 & & \mid & \exists X.A \\
 & & \mid & U\underline{B} \\
\text{Computation Types} & \underline{B} & ::= & A \to \underline{B} \\
 & & \mid & A \mathrel{-\!\!*} \underline{B} \\
 & & \mid & \forall X.\underline{B} \\
 & & \mid & FA \\
\text{Values} & V & ::= & x \\
 & & \mid & tt \\
 & & \mid & \sigma \\
 & & \mid & \text{inj}_V V \\
 & & \mid & (V, V) \\
 & & \mid & (V * V) \\
 & & \mid & \text{pack } (A, V) \text{ as } \exists X.A \\
 & & \mid & \text{thunk } M \\
\text{Computations} & M & ::= & \lambda x\colon A.M \\
 & & \mid & MV \\
 & & \mid & \alpha x\colon A.M \\
 & & \mid & M@V \\
 & & \mid & \Lambda X.M \\
 & & \mid & M[A] \\
 & & \mid & \text{ret } V \\
 & & \mid & \text{force } V \\
 & & \mid & \text{newcase}_A x; M \\
 & & \mid & \text{match V with V } \{ \text{ inj } x.M \| N \ \} \\
 & & \mid & \text{let } (x, x) = V; M \\
 & & \mid & \text{let } (x * x) = V; M \\
 & & \mid & \text{unpack } (X, x) = V; M \\
\text{Value Context} & \Gamma & ::= & \cdot \\
 & & \mid & \Gamma, x\colon A \\
 & & \mid & \Gamma * x\colon A \\
\text{Type Context} & \Delta & ::= & \cdot \\
 & & \mid & \Delta, X
\end{array}
$$

1.2 Typed Terms

$$\frac{}{\Delta;\Gamma, x\colon A \vdash_v x : A}$$

$$\frac{}{\Delta;\Gamma * x\colon A \vdash_v x : A}$$

$$\frac{}{\Delta;\Gamma \vdash_v \mathrm{tt} : \mathrm{Unit}}$$

$$\frac{\Delta;\Gamma \vdash_v \sigma : \mathrm{Case}A \qquad \Delta;\Gamma \vdash_v V : A}{\Delta;\Gamma \vdash_v \mathrm{inj}_\sigma V : \mathrm{OSum}}$$

$$\frac{\Delta;\Gamma \vdash_v V_1 : A_1 \qquad \Delta;\Gamma \vdash_v V_2 : A_2}{\Delta;\Gamma \vdash_v (V_1, V_2) : A_1 \times A_2}$$

$$\frac{\Delta;\Gamma_1 \vdash_v V_1 : A_1 \qquad \Delta;\Gamma_2 \vdash_v V_2 : A_2}{\Delta;\Gamma_1 * \Gamma_2 \vdash_v (V_1 * V_2) : A_1 * A_2}$$

$$\frac{\Delta;\Gamma \vdash_v V : A[A'/X]}{\Delta;\Gamma \vdash_v \mathrm{pack}(A', V) \text{ as } \exists X.A : \exists X.A}$$

$$\frac{\Delta;\Gamma \vdash_c M : \underline{B}}{\Delta;\Gamma \vdash_v \mathrm{thunk}\ M : U\underline{B}}$$

$$\frac{\Delta;\Gamma, x : A \vdash_c M : \underline{B}}{\Delta;\Gamma \vdash_c \lambda x : A.M : A \to \underline{B}}$$

$$\frac{\Delta;\Gamma \vdash_c M : A \to \underline{B} \qquad \Delta;\Gamma \vdash_v N : A}{\Delta;\Gamma \vdash_c MN : \underline{B}}$$

$$\frac{\Delta;\Gamma * x : A \vdash_c M : \underline{B}}{\Delta;\Gamma \vdash_c \alpha x : A.M : A \mathrel{-\!\!*} \underline{B}}$$

$$\frac{\Delta;\Gamma_1 \vdash_c M : A \mathrel{-\!\!*} \underline{B} \qquad \Delta;\Gamma_2 \vdash_v N : A}{\Delta;\Gamma_1 * \Gamma_2 \vdash_c M@N : \underline{B}}$$

$$\frac{\Delta, X;\Gamma \vdash_c M : \underline{B}}{\Delta;\Gamma \vdash_c \Lambda X.M : \forall X.\underline{B}}$$

$$\frac{\Delta;\Gamma \vdash_c\colon M : \forall X.\underline{B} \qquad \Delta \vdash A}{\Delta;\Gamma \vdash_c M[A] : \underline{B}[A/X]}$$

$$\frac{\Delta;\Gamma \vdash_v V : A}{\Delta;\Gamma \vdash_c \mathrm{ret}\ V : FA}$$

$$\frac{\Delta;\Gamma\vdash_v V : U\underline{B}}{\Delta;\Gamma\vdash_c\text{: force } V : \underline{B}}$$

$$\frac{\Delta;\Gamma * (\sigma : \text{Case}A)\vdash_c M : \underline{B} \qquad \Delta\vdash A}{\Delta;\Gamma\vdash_c \text{newcase}_A x; M : \underline{B}}$$

$$\frac{\Delta;\Gamma\vdash_v V : \text{OSum} \qquad \Delta;\Gamma\vdash_v \sigma : \text{Case } A \qquad \Delta;\Gamma, x : A\vdash M : \underline{B} \qquad \Delta;\Gamma\vdash_c N : \underline{B}}{\Delta;\Gamma\vdash_c \text{match } V \text{ with } \sigma\{ \text{ inj } x.M\| N\} : \underline{B}}$$

$$\frac{\Delta;\Gamma\vdash_v V : A_1\times A_2 \qquad \Delta;\Gamma, x : A_1, y : A_2\vdash_c M : \underline{B}}{\Delta;\Gamma\vdash_c \text{let } (x, y) = V; M : \underline{B}}$$

$$\frac{\Delta;\Gamma\vdash_v V : A_1 * A_2 \qquad \Delta;\Gamma * x : A_1 * y : A_2\vdash_c M : \underline{B}}{\Delta;\Gamma\vdash_c \text{let } (x * y) = V; M : \underline{B}}$$

$$\frac{\Delta\vdash \underline{B} \qquad \Delta;\Gamma\vdash_v V : \exists X.A \qquad \Delta, X;\Gamma, x : A\vdash_c M : \underline{B}}{\Delta;\Gamma\vdash_c\text{: unpack}(X, x) = V; M : \underline{B}}$$

2 Meta Language

2.1 Raw Formulas

Formula	ϕ, ψ	::=	$t =_A u$
		$\mid$	$R(t, u)$
		$\mid$	$\phi \implies \psi$
		$\mid$	$\phi\wedge\psi$
		$\mid$	$\phi\vee\psi$
		$\mid$	$\exists x\colon A.\phi$
		$\mid$	$\exists X.\phi$
		$\mid$	$\exists \underline{X}.\phi$
		$\mid$	$\exists R\colon \mathit{Rel}[A, B].\phi$
		$\mid$	$\forall x\colon A.\phi$
		$\mid$	$\forall X.\phi$
		$\mid$	$\forall \underline{X}.\phi$
		$\mid$	$\forall R\colon \mathit{Rel}[A, B].\phi$

2.2 Typed Formulas

Propositions, or well-formed formulas, use a term environment Γ, type environment Δ and relation environment Θ. The typing judgement for Propositions is $\Delta;\Gamma;\Theta\vdash_p P$. There are value relations and computation relations.

$$\frac{\Delta;\Gamma\vdash_{\_} t : A \qquad \Delta;\Gamma\vdash_{\_} u : A}{\Delta;\Gamma;\Theta\vdash_p t =_A u}\text{ for } v, c$$

$$\frac{\Delta;\Gamma\vdash_{\_} t : A \qquad \Delta;\Gamma\vdash_{\_} u : B \qquad R : \mathit{Rel}_{\_}[A, B]\in\Theta}{\Delta;\Gamma;\Theta\vdash_p R(t, u)}\text{ for } v, c$$

$$\frac{\Delta;\Gamma;\Theta \vdash_p \phi \qquad \Delta;\Gamma;\Theta \vdash_p \psi}{\Delta;\Gamma;\Theta \vdash_p \phi \Box \psi}\ \Box \in \{\wedge, \vee, \implies\}$$

what about *? Something like exists fresh

$$\frac{\Delta;\Gamma, x : A|\Theta \vdash_p \phi}{\Delta;\Gamma;\Theta \vdash_p \exists x : A.\phi}$$

$$\frac{\Delta;\Gamma;\Theta \vdash_p \phi}{\Delta;\Gamma;\Theta \vdash_p \exists X.\phi}\ X \notin FV(\Delta,\Gamma,\Theta)$$

$$\frac{\Delta;\Gamma;\Theta \vdash_p \phi}{\Delta;\Gamma;\Theta \vdash_p \exists \underline{X}.\phi}\ \underline{X} \notin FV(\Delta,\Gamma,\Theta)$$

$$\frac{\Delta;\Gamma;\Theta, R : Rel\_[A,B] \vdash_p \phi}{\Delta;\Gamma;\Theta \vdash_p \exists R : Rel\_[A,B].\phi}\ \text{for } \{v, c\}$$

$$\frac{\Delta;\Gamma, x : A|\Theta \vdash_p \phi}{\Delta;\Gamma;\Theta \vdash_p \forall x : A.\phi}$$

$$\frac{\Delta;\Gamma;\Theta \vdash_p \phi}{\Delta;\Gamma;\Theta \vdash_p \forall X.\phi}\ X \notin FV(\Delta,\Gamma,\Theta)$$

$$\frac{\Delta;\Gamma;\Theta \vdash_p \phi}{\Delta;\Gamma;\Theta \vdash_p \forall \underline{X}.\phi}\ \underline{X} \notin FV(\Delta,\Gamma,\Theta)$$

$$\frac{\Delta;\Gamma;\Theta, R : Rel\_[A,B] \vdash_p \phi}{\Delta;\Gamma;\Theta \vdash_p \forall R : Rel\_[A,B].\phi}\ \text{for } \{v, c\}$$

2.3 Typed Relations

Relations are of the form $(x : A, y : B).\phi$ where ϕ is a proposition that can use x, y. The typing judgement for relations is $\Delta;\Gamma;\Theta \vdash_r (x : A, y : B).\phi : Rel\_[A,B]$. The body of the relation is a proposition. Here we pay attention to the difference between value and computation relations.

again, what about *?

$$\frac{\Delta;\Gamma, x : A \vdash_v t : C \qquad \Delta;\Gamma, y : B \vdash_v u : C}{\Delta;\Gamma;\Theta \vdash_r (x : A, y : B).t =_C u : Rel_v[A,B]}$$

secretly inserting stoup

$$\frac{\Delta;\Gamma|x : \underline{A} \vdash_c t : \underline{C} \qquad \Delta;\Gamma|y : \underline{B} \vdash_c u : \underline{C}}{\Delta;\Gamma;\Theta \vdash_r (x : \underline{A}, y : \underline{B}).t =_{\underline{C}} u : Rel_c[\underline{A},\underline{B}]}$$

Given some $x : A$ and $y : B$ the terms t, u are related by R, thus we have a relation on A, B. Think of these like a lambda abstraction over two parameters. If the body is related, we can

$$\frac{\Delta;\Gamma,x:A\vdash_v t:C \qquad \Delta;\Gamma,y:B\vdash_v u:D}{\Delta;\Gamma;\Theta,R:Rel_v[C,D]\vdash_r (x:A,y:B).R(t,u):Rel_v[A,B]}$$

$$\frac{\Delta;\Gamma|x:\underline{A}\vdash_c t:\underline{C} \qquad \Delta;\Gamma|y:\underline{B}\vdash_c u:\underline{D}}{\Delta;\Gamma;\Theta,\underline{R}:Rel_c[\underline{C},\underline{D}]\vdash_r (x:\underline{A},y:\underline{B}).\underline{R}(t,u):Rel_c[\underline{A},\underline{B}]}$$

What is the intuition here? This rule is in figure 5 of the PE logic paper.

$$\frac{\Delta;\Gamma;\Theta\vdash_p \phi \qquad \Delta;\Gamma;\Theta\vdash_r (x:A,y:B).\psi:Rel_vA,B}{\Delta;\Gamma;\Theta\vdash_r (x:A,y:B).\phi\implies\psi:Rel_v[A,B]}\text{ also } c \text{ version}$$

?

$$\frac{\Delta;\Gamma;\Theta\vdash ? \qquad \Delta;\Gamma;\Theta\vdash ?}{\Delta;\Gamma;\Theta\vdash_r (x:A,y:B).\phi\wedge\psi:Rel_v[A,B]}\text{ also } c \text{ version}$$

$$\frac{\Delta;\Gamma,z:C;\Theta\vdash_r (x:A,y:B).\phi:Rel_v[A,B]?}{\Delta;\Gamma;\Theta\vdash_r (x:A,y:B).\forall(z:C).\phi:Rel_v[A,B]}\text{ also } c \text{ version}$$

$$\frac{\Delta,X;\Gamma;\Theta\vdash_r (x:A,y:B).\phi:Rel_v[A,B]?}{\Delta;\Gamma;\Theta\vdash_r (x:A,y:B).\forall X.\phi:Rel_v[A,B]}\text{ also } c \text{ version}$$

$$\frac{\Delta,X;\Gamma;\Theta,R:Rel_{\mathbf{n}}[C,D]\vdash_r (x:A,y:B).\phi:Rel_{\mathbf{m}}[A,B]}{\Delta;\Gamma;\Theta\vdash_r (x:A,y:B).\forall(R:Rel_{\mathbf{n}}[C,D]).\phi:Rel_{\mathbf{m}}[A,B]}\ \mathbf{n},\mathbf{m}\in\{v,c\}$$

Analogous versions for $\exists$ connectives.

2.4 Deduction Rules

The judgement for deduction sequence are of the form $\Delta;\Gamma;\Theta;\Phi\vdash_d\psi$ where Δ is a type environment, Γ is a term environment, Θ is a relation environment, Θ is a proposition environment, and ψ is a proposition. like term intro and elim, but without proof terms also for computations?

$$\frac{\Delta;\Gamma\vdash_v t:A}{\Delta;\Gamma;\Theta;\Phi\vdash_d t=_A t}\text{ refl}$$

$$\frac{\Delta;\Gamma;\Theta;\Phi\vdash_d t=_A u \qquad \Delta;\Gamma;\Theta;\Phi\vdash_d \phi[t/x]}{\Delta;\Gamma;\Theta;\Phi\vdash_d \phi[u/x]}\text{ subst}$$

$$\frac{\Delta;\Gamma;\Theta;\Phi,\phi\vdash_d\psi}{\Delta;\Gamma;\Theta;\Phi\vdash_d\phi\implies\psi}\implies\text{Intro}$$

$$\frac{\Delta; \Gamma; \Theta; \Phi \vdash_d \phi \implies \psi \qquad \Delta; \Gamma; \Theta; \Phi \vdash_d \phi}{\Delta; \Gamma; \Theta; \Phi \vdash_d \psi} \implies \text{Elim}$$

and familiar rules for logical and ($\wedge$)

$$\frac{\Delta; \Gamma, (x : A); \Theta; \Phi \vdash_d \phi}{\Delta; \Gamma; \Theta; \Phi \vdash_d \forall (x : A).\phi} \forall \text{ Term Intro}$$

$$\frac{\Delta; \Gamma; \Theta; \Phi \vdash_d \forall (x : A).\phi \qquad \Delta; \Gamma \vdash_v t : A}{\Delta; \Gamma; \Theta; \Phi \vdash_d \phi[t/x]} \forall \text{ Term Elim}$$

$$X \notin FV(..), \text{ also for } c \ \frac{\Delta, X; \Gamma; \Theta; \Phi \vdash_d \phi}{\Delta; \Gamma; \Theta; \Phi \vdash_d \forall X.\phi} \forall \text{ Type Intro}$$

$$\text{also for } c \ \frac{\Delta; \Gamma; \Theta; \Phi \vdash_d \forall X.\phi \qquad \Delta \vdash A}{\Delta; \Gamma; \Theta; \Phi \vdash_d \phi[A/X]} \forall \text{ Type Elim}$$

$$\text{also for } c \ \frac{\Delta; \Gamma; \Theta, R : Rel_v[A, B]; \Phi \vdash_d \phi}{\Delta; \Gamma; \Theta; \Phi \vdash_d \forall (R : Rel_v[A, B]).\phi} \forall \text{ Rel Intro}$$

$$\text{also for } c \ \frac{\Delta; \Gamma; \Theta; \Phi \vdash_d \forall (R : Rel_v[A, B]).\phi \qquad \Delta; \Gamma; \Theta \vdash_r (x : A, y : B).\psi : Rel_v[A, B]}{\Delta; \Gamma; \Theta; \Phi \vdash_d \phi[\psi[t/x, u/y]/R(t, u)]} \forall \text{ Rel Elim}$$

2.5 Axioms & Axiom Schemas

2.5.1 Congruences

$$\frac{\Delta; \Gamma, (x : A) \vdash_c t, u : \underline{B} \qquad \Delta; \Gamma, (x : A); \Theta; \Phi \vdash_d t = u}{\Delta; \Gamma; \Theta; \Phi \vdash_d (\lambda (x : A).t) =_{A \to \underline{B}} (\lambda (x : A).u)} \lambda \text{ cong}, x \notin FV(\Phi)$$

2.5.2 Beta / Eta Laws

$$\frac{}{\Delta; \Gamma; \Theta; \Phi \vdash_d \forall (u : A).((\lambda x : A.t) u =_B t[u/x])} \lambda\beta$$

$$\frac{}{\Delta; \Gamma; \Theta; \Phi \vdash_d \forall X.\forall Y.\forall (f : X \to Y).((\lambda x : X.fx) =_{X \to Y} f)} \lambda\eta$$

2.5.3 Parametricity

2.6 Example Derivations

2.6.1 Equality Reasoning

Transitivity

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{}{...; (x =_X y) \wedge (y =_X z) \vdash_d (x =_X y) \wedge (y =_X z)}
}{...; (x =_X y) \wedge (y =_X z) \vdash_d y =_X z}
\qquad
\dfrac{
\dfrac{
\dfrac{}{...; (x =_X y) \wedge (y =_X z) \vdash_d (x =_X y) \wedge (y =_X z)}
}{...; (x =_X y) \wedge (y =_X z) \vdash_d (x =_X y)}
}{...; (x =_X y) \wedge (y =_X z) \vdash_d (x =_X y)[y/y]}
}{X; x, y, z; \cdot; (x =_X y) \wedge (y =_X z) \vdash_d (x =_X y)[z/y]}\,\text{subst}
}{X; x, y, z; \cdot; (x =_X y) \wedge (y =_X z) \vdash_d (x =_X z)}
}{\vdash_d \forall X. \forall (xyz : X).(x =_X y) \wedge (y =_X z) \implies (x =_X z)}
$$

Symmetry Use IdExt and opRel?

2.6.2 Extensionality

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{}{X, \underline{Y}; f, g, x; \cdot; (\forall (x : X). fx =_{\underline{Y}} gx) \vdash_d \forall (x : X).(fx =_{\underline{Y}} gx)}\,\text{in } \Phi
\qquad
\dfrac{}{... \vdash_v x : X}\,\text{Var}
}{X, \underline{Y}; f, g, x; \cdot; (\forall (x : X). fx =_{\underline{Y}} gx) \vdash_d (fx =_{\underline{Y}} gx)}\,\forall\ \text{Elim}
}{X, \underline{Y}; f, g; \cdot; (\forall (x : X). fx =_{\underline{Y}} gx) \vdash_d ((\lambda x : X. fx) =_{X \to \underline{Y}} (\lambda x : X. gx))}\,\lambda\ \text{cong}
}{X, \underline{Y}; f, g; \cdot; (\forall (x : X). fx =_{\underline{Y}} gx) \vdash_d (f =_{X \to \underline{Y}} g)}\,\lambda\eta
}{X, \underline{Y}; f, g; \cdot; \cdot \vdash_d ((\forall (x : X). fx =_{\underline{Y}} gx) \implies f =_{X \to \underline{Y}} g)}\,\text{Intros}
}{... \vdash_d \forall (f, g : X \to \underline{Y}).((\forall (x : X). fx =_{\underline{Y}} gx) \implies f =_{X \to \underline{Y}} g)}\,\text{Intros}
$$

2.6.3 Identity Extension Lemma

By Induction on Types.

For Unit Recall

$$eq_A := (x : A, y : A). x =_A y$$

and substitution of a relation into a base type is just the relational interpretation of the base type.

$$Unit[eq_{Unit}] = [\![Unit]\!]_{Rel} = \{(tt, tt)\}$$

What rules are missing here to make this proof go through?

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{}{...; \Phi, \{u = tt, v = tt\} \vdash_d tt =_{Unit} tt}\,\text{Refl}
}{...; \Phi, u([\![Unit]\!]_{Rel})v \vdash_d u =_{Unit} v}\,\text{Unit } \eta\ ?
}{... \vdash_d u([\![Unit]\!]_{Rel})v \implies u =_{Unit} v}\,\text{Intro}
\qquad
\dfrac{
\dfrac{
\dfrac{}{...; \Phi, tt =_{Unit} tt \vdash_d tt([\![Unit]\!]_{Rel})tt}\,\text{By Def}
}{...; \Phi, u =_{Unit} v \vdash_d u([\![Unit]\!]_{Rel})v}\,\text{Unit } \eta?
}{... \vdash_d u =_{Unit} v \implies u([\![Unit]\!]_{Rel})v}\,\text{Intro}
}{\cdot; u : Unit, v : Unit; \cdot; \cdot \vdash_d u([\![Unit]\!]_{Rel})v \equiv u =_{Unit} v}\,\text{Logical Equiv}
}{\cdot; \cdot; \cdot; \cdot \vdash_d \forall u : Unit, v : Unit. u([\![Unit]\!]_{Rel})v \equiv u =_{Unit} v}\,\text{Intros}
}{\cdot; \cdot; \cdot; \cdot \vdash_d \forall u : Unit, v : Unit. u(Unit[eq_{Unit}])v \equiv u =_{Unit} v}\,\text{Relational Subst}
$$

For $X \to \underline{Y}$ Recall

$$\begin{aligned}(X \to \underline{Y})[eq_X, eq_{\underline{Y}}] = (f : (X \to \underline{Y}) g : (X \to \underline{Y})).\\ \forall (x : X). \forall (x' : X).\\ x (eq_X) x' \implies (fx)(eq_Y)(gx')\end{aligned}$$

One direction.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\text{(the relation in above on } f, g)}\text{ in }\Phi \quad \dfrac{}{.. \vdash_v x : X}\text{ Var}}{.. \vdash_d \forall (x' : X). x(eq_X)x' \implies fx =_{\underline{Y}} gx'}\ \forall\text{ Elim} \quad \dfrac{}{.. \vdash_v x : X}\text{ Var}}{..; f((X \to \underline{Y})[eq_X, eq_{\underline{Y}}])g \vdash_d x(eq_X)x \implies fx =_{\underline{Y}} gx}\ \forall\text{ Elim} \quad \dfrac{\dfrac{}{\vdash_d x =_X x}\text{ Refl}}{\vdash_d x(eq_X)x}\text{ IdExt}}{..; f, g, x; ..; f((X \to \underline{Y})[eq_X, eq_{\underline{Y}}])g \vdash_d fx(eq_{\underline{Y}})gx} \implies \text{Elim}}{..; f, g, x; ..; f((X \to \underline{Y})[eq_X, eq_{\underline{Y}}])g \vdash_d fx =_{\underline{Y}} gx}\text{ IdExt}}{..; f, g; ..; f((X \to \underline{Y})[eq_X, eq_{\underline{Y}}])g \vdash_d (\lambda x : X. fx) =_{X \to \underline{Y}} (\lambda x : X. gx)}\ \lambda\text{ cong}}{..; f, g; ..; f((X \to \underline{Y})[eq_X, eq_{\underline{Y}}])g \vdash_d f =_{X \to \underline{Y}} g}\ \lambda\eta}{..; f, g; .. \vdash_d f((X \to \underline{Y})[eq_X, eq_{\underline{Y}}])g \implies f =_{X \to \underline{Y}} g}\text{ Intros}}{... \vdash_d \forall (f, g : X \to \underline{Y}). f((X \to \underline{Y})[eq_X, eq_{\underline{Y}}])g \implies f =_{X \to \underline{Y}} g}\text{ Intros}$$

2.6.4 Identity Function

$$\dfrac{\dfrac{?}{X; f : \forall X. X \to FX, x : X; \cdot; \cdot \vdash_d f[X]x =_{FX} ret\ x}}{\cdot; f : \forall X. X \to FX; \cdot; \cdot \vdash_d \forall X. \forall (x : X). f[X]x =_{FX} ret\ x}\text{ Intro}$$

2.6.5 Church Encodings

Unit Forget the built in Unit type for now. Note that we only have the compuational function type in this language.

$$\begin{aligned}&Unit := \forall X. X \to FX\\ &\mathbf{1} : Unit\\ &\mathbf{1} = \Lambda X. \lambda (x : X).\ \text{ret } x\end{aligned}$$