1 Object Language

1.1 Raw Terms

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

Value Types	A	$::=$	X
		$\mid$	Unit
		$\mid$	Case A
		$\mid$	OSum
		$\mid$	$A \times A$
		$\mid$	$A * A$
		$\mid$	$\exists X.A$
		$\mid$	$U\underline{B}$
Computation Types	$\underline{B}$	$::=$	$A \to \underline{B}$
		$\mid$	$A \twoheadrightarrow* \underline{B}$
		$\mid$	$\forall X.\underline{B}$
		$\mid$	FA
Values	V	$::=$	x
		$\mid$	tt
		$\mid$	σ
		$\mid$	$\mathrm{inj}_V V$
		$\mid$	(V, V)
		$\mid$	$(V * V)$
		$\mid$	pack (A, V) as $\exists X.A$
		$\mid$	thunk M
Computations	M	$::=$	$\lambda x\colon A.M$
		$\mid$	MV
		$\mid$	$\alpha x\colon A.M$
		$\mid$	$M@V$
		$\mid$	$\Lambda X.M$
		$\mid$	$M[A]$
		$\mid$	ret V
		$\mid$	force V
		$\mid$	$\mathrm{newcase}_A x; M$
		$\mid$	match V with V { inj $x.M \| N$ }
		$\mid$	let $(x, x) = V; M$
		$\mid$	let $(x * x) = V; M$
		$\mid$	unpack $(X, x) = V; M$
Value Context	Γ	$::=$	$\cdot$
		$\mid$	$\Gamma, x\colon A$
		$\mid$	$\Gamma * x\colon A$
Type Context	Δ	$::=$	$\cdot$
		$\mid$	Δ, X

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

$$\begin{array}{llll}
\text{Formula} & \phi, \psi & ::= & t =_A u \\
 & & | & R(t, u) \\
 & & | & \phi \implies \psi \\
 & & | & \phi \wedge \psi \\
 & & | & \phi \vee \psi \\
 & & | & \exists x \colon A.\phi \\
 & & | & \exists X.\phi \\
 & & | & \exists \underline{X}.\phi \\
 & & | & \exists R \colon \mathit{Rel}[A, B].\phi \\
 & & | & \forall x \colon A.\phi \\
 & & | & \forall X.\phi \\
 & & | & \forall \underline{X}.\phi \\
 & & | & \forall R \colon \mathit{Rel}[A, B].\phi
\end{array}$$

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

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

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

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

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

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

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

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

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

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 $\Delta;\Gamma;\Theta \vdash_r (x : A, y : B).\phi : \mathit{Rel}\_[A,B]$. The body of the relation is a proposition. Here we pay attention to the difference between value and computation relations.

again, what about *?

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

secretly inserting stoup

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

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.

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

?

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

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

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

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

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

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

Symmetry Use IdExt and opRel?

2.6.2 Extensionality

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

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.

$$Unit[eq_{Unit}] = \llbracket Unit \rrbracket_{Rel} = \{(tt, tt)\}$$

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

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

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.

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

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.

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

Given

$$f : \forall X.X \to FX$$

by η

$$f = \Lambda X.\lambda(x : X).f[X]x$$

Pick a relation on FX,

$$R := X; x : X; \cdot \vdash_r (a : FX, b : FX).\ \text{ret}\ x =_{FX} b$$

What does parametricity say for the type $\forall X.X \to FX$?

$$\forall (t : (\forall X, X \to FX)).\forall Y, Z, R : Rel_v[Y, Z].t[Y]((X \to FX)[R])t[Z]$$

$((X \to FX)[R])$ Substitute the relation in for type variables.

$$\begin{aligned} &(X \to FX)[R] = \\ &= X[R] \to (FX)[R] \\ &= X[R] \to {\color{red}?} \end{aligned}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{?}{Unit, X; \mathbf{1}, f, x; \cdot; \cdot \vdash_d \forall X.\forall(x : X).\ \text{ret}\ x =_{FX}\ f[X]x} \qquad \dfrac{}{Unit, X \vdash X}}{Unit, X; \mathbf{1}, f, x; \cdot; \cdot \vdash_d \forall(x : X).\ \text{ret}\ x =_{FX}\ f[X]x}\ \forall\ \text{Elim} \qquad \dfrac{}{.. \vdash_v x : X}\ \text{Var}}{Unit, X; \mathbf{1}, f, x; \cdot; \cdot \vdash_d\ \text{ret}\ x =_{FX}\ f[X]x}\ \forall\ \text{Elim}}{Unit; \mathbf{1}, f; \cdot; \cdot \vdash_d \Lambda X.\lambda(x : X).\ \text{ret}\ x =_{Unit} \Lambda X.\lambda(x : X).f[X]x}\ \text{cong}}{Unit; \mathbf{1}, f; \cdot; \cdot \vdash_d \mathbf{1} =_{Unit} f}\ \text{Def}, \eta}{Unit; \mathbf{1}; \cdot; \cdot \vdash_d \forall(f : Unit).\mathbf{1} =_{Unit} f}\ \text{Intro}$$

2.6.6 OSum Free Theorems

Looking for free theorems for types containing our separating connectives, OSum, and Case. From the gradual parametricity paper: given

$$\vdash M : \forall^{\nu} X.? \to X$$

and $\vdash V : ?$ then

$$\text{unseal}_X(M\{X \cong A\}V)\text{true}$$

either diverges or errors.
In CBPV OSum:

$$\vdash_c M : \forall X.\text{Case}X \twoheadrightarrow\!\!\!\!* \ (\text{OSum} \to FX)$$

should be uninhabited. If error $\mho$ was added to the language, then

$$A; \sigma : \text{Case}A * d : \text{OSum} \vdash_c (M[A]@\sigma)d : FX$$

should always error.

$$A; \sigma : \mathrm{Case} A{*}d : \mathrm{OSum}; \cdot; \cdot \vdash_d \forall (M : X.\mathrm{Case} X \mathrel{-\!\!*} (\mathrm{OSum} \to FX)).(M[A]@\sigma)d =_{FX} \mho$$

Seems like we have to state this property with the context loaded since we dont have a freshenss quantifier. Would we want something like?:

$$\vdash_d \forall A.\text{И}(\sigma : \text{Case A}).\forall (d : \mathrm{OSum}) \forall (M : X.\mathrm{Case} X \mathrel{-\!\!*} (\mathrm{OSum} \to FX)).(M[A]@\sigma)d =_{FX} \mho$$