Let A, B, C types. What is the relationship between

$$(A \times B \to C) \quad \text{and} \quad (A \to B \to C) \;?$$

We can exhibit a logical equivalence:

$$\text{ev-pair}: (A \times B \to C) \longrightarrow (A \to B \to C)$$

defined by $\quad g \longmapsto \quad a \mapsto b \mapsto g(a,b)$

Note this definition uses the intro rule for product types

$$\langle -,- \rangle : A \to B \to A \times B$$

We also require

$$\text{ind}_\times : (A \to B \to C) \longrightarrow (A \times B \to C)$$

and this comes from the elimination rule for product types which gives the following map. For any $p : A \times B \vdash P(p)$

$$\text{ind}_\times : \prod_{a:A} \prod_{b:B} P(a,b) \longrightarrow \prod_{p:A\times B} P(p)$$

Q: Can we promote this to an equivalence?

This requires homotopies $\quad \text{ev-pair} \circ \text{ind}_\times \sim \text{id}$

Ie for each $f : A \to B \to C$ we need an identification

$$\text{ev-pair}(\text{ind}_\times f) =_{A \to B \to C} f$$

For each $a:A$ and $b:B$ we have definitional equalities

$$(\text{ev-pair}(\text{ind}_\times f))(a,b) \doteq (\text{ind}_\times f)(a,b) \doteq f(a,b)$$

where this last definitional equality is the computation rule for the product type.

By the η-rule for function types $f : A \to B \to C$

$$f \doteq \lambda a. \lambda b. f(a,b)$$

By above we calculated $\text{ev-pair}(\text{ind}_\times f)(a,b) \doteq f(a,b)$.

$$\text{ev-pair}(\text{ind}_\times f) = \lambda a. \lambda b. f(a,b).$$

Thus $\text{refl} : \text{ev-pair}(\text{ind}_\times f) =_{A \to B \to C} f$

This gives the homotopy $\text{ev-pair} \circ \text{ind}_\times \sim \text{id}$.

The final step requires a homotopy $\text{ind}_\times \circ \text{ev-pair} \sim \text{id}$

So we require identifications for all $g : A \times B \to C$

$$\text{ind}_\times(\text{ev-pair}\ g) =_{A \times B \to C} g$$

$$\text{ind}_\times(\text{ev-pair}\ g) \doteq \text{ind}_\times(\lambda a \lambda b. g(a,b))$$

So for $a : A$ and $b : B$ we have

$$(\text{ind}_\times(\text{ev-pair}\ g))(a,b) \doteq g(a,b)$$

$$\lambda a \lambda b. \text{refl} : \prod_{a:A} \prod_{b:B} (\text{ind}_\times(\text{ev-pair}\ g))(a,b) = g(a,b))$$

Using the induction for product types we can promote this

$$\prod_{a:A} \prod_{b:B} (\text{ind}_\times(\text{ev-pair}\ g))(a,b) = g(a,b) \xrightarrow{\text{ind}_\times} \prod_{p : A \times B} (\text{ind}_\times(\text{ev-pair}\ g))\, p = g(p)$$

$$\text{ind}_\times(\lambda a. \lambda b. \text{refl}) : \prod_{p : A \times B} \text{ind}_\times(\text{ev-pair}\ g)(p) = g(p)$$

$$\text{ind}_\times(\lambda a \lambda b. \text{refl}) : \text{ind}_\times(\text{ev-pair}\ g) \sim g$$

We're stuck here b/c we cannot turn this homotopy into an identification

$$? : \text{idx}(\text{ev-pair } g) =_{A \times B \to C} g.$$

The function extensionality axiom characterizes identity types of function types.

=

§ Function extensionality

This characterizes identity types of function types.

It's an axiom that we will add to Martin-Löf type theory.

Prop (function extensionality) For a type family $B : A \to \mathcal{U}$ the following are logically equivalent:

(1) The family of maps $\qquad (f = g) \simeq (f \sim g)$

$$\text{htpy-id} : \prod_{f,g : \prod_{x:A} B(x)} (f = g) \to (f \sim g)$$

defined by $\text{refl}_f \longmapsto \text{refl-htpy}_f$ is a family of equivalences

(2) For any $f : \prod_{x:A} B(x)$ the type

$$\sum_{g : \prod_{x:A} B(x)} f \sim g \quad \text{is contractible onto } (f, \text{refl-htpy}_f).$$

(3) The principle of homotopy induction holds: for any P depending on $f, g : \prod_{x:A} B(x)$ and $H : f \sim g$ the evaluation map

$$\text{ev} : \left(\prod_{f,g : \prod_{x:A} B(x)} \prod_{H : f \sim g} P(f, g, H) \right) \to \left(\prod_{f : \prod_{x:A} B(x)} P(f, f, \text{refl-htpy}) \right)$$

has a section.

proof: This is a specialization of the universal property of identity types.

There is a fourth equivalent characterization

Theorem For any universe $\mathcal{U}$ the following are logically equivalent

(i) the function extensionality principle holds in $\mathcal{U}$: for any type family $B : A \to \mathcal{U}$ the map

$$\text{htpy-id} : \prod_{f,g : \prod_{x:A} B(x)} (f = g) \to f \sim g)$$ is a family of equivalences.

(ii) The weak function extensionality principle holds: for any $B : A \to \mathcal{U}$

$$\left(\prod_{x:A} \text{is-contr}(B(x))\right) \longrightarrow \text{is-contr}\left(\prod_{x:A} B(x)\right).$$

Proof: Assume (i). Consider $B : A \to \mathcal{U}$ and suppose for all $x : A$ $B(x)$ is contractible w/ $c(x) : B(x)$ as center of contraction $h(x) : \prod_{y : B(x)} c(x) = y$ as contracting homotopy. Define

$c := \lambda x.\, c(x) : \prod_{x:A} B(x)$ to be the center of contraction.

Need a homotopy $\prod_{f : \prod_{x:A} B(x)} c = f$. By fun ext it suffices to give a term of type $(c \sim f) := \prod_{x:A} c(x) = f(x)$

This is given by $\lambda x.\, h(x, f(x)) : c \sim f$.

Assume (ii). We must show $\displaystyle\sum_{g:\prod_{x:A} B(x)} f \sim g$ is contractible.

Consider the section-retraction pair

$$\left(\sum_{g:\prod_{x:A} B(x)} f \sim g\right) \xrightarrow{s} \left(\prod_{x:A} \sum_{y:B(x)} f(x) = y\right) \xrightarrow{r} \left(\sum_{g:\prod_{x:A} B(x)} f \sim g\right)$$

where
$$s := \lambda(g,H).\lambda x.(g(x), H(x))$$
$$r := \lambda p.(\lambda x.\mathrm{pr}_1 p(x), \lambda x.\mathrm{pr}_2 p(x))$$

The composite $r \circ s$ is homotopic to id by computation rules for Σ and Π types. The middle type is a product of a family of contractible types. By weak fun ext the product is contractible. Since retracts of contractible types are contractible we're done. ∎

=

The rest of today will be applications.

Theorem For any $B : A \to \mathcal{U}$

$$\prod_{x:A} \text{is-trunc}_k(B(x)) \longrightarrow \text{is-trunc}_k\left(\prod_{x:A} B(x)\right)$$

proof: Base case where $k=-2$ is weak fun ext.
For the inductive step we assume that products of families of k-types are k-types. Want to show that if $B : A \to \mathcal{U}$ and $\prod_{x:A} B(x)$ is a $k+1$-type then $\prod_{x:A} B(x)$ is a $k+1$-type.

We must show then for all $f, g : \prod_{x:A} B(x)$ that $f = g$ is a k-type.

This is equivalent to $(f \sim g) := \prod_{x:A} f(x) =_{B(x)} g(x)$ by funext.

Since $B(x)$ is a k+1 type, $f(x) =_{B(x)} g(x)$ is a k-type. By the IH

$f \sim g$ is a k-type. $\square$.

Cor Let B be a k-type and let A be any type. Then

$A \to B$ is a k-type.

proof. This is a special case of the previous

Cor For any type A, $(\neg A) := A \to \emptyset$ is a proposition.

Exercise Prove other types are propositions.

§ the type theoretic principle of choice

This asserts the fact that $\prod$ types distribute over $\sum$ types.

Theorem For any family of types $x:A, y:B(x) \vdash C(x,y)$ type the map

$$\text{choice} : \left(\prod_{x:A} \sum_{y:B(x)} C(x,y)\right) \to \left(\sum_{f:\prod_{x:A} B(x)} \prod_{x:A} C(x, f(x))\right)$$

defined by

$$\text{choice}(h) := (\lambda x.\, pr_1(h(x)),\ \lambda x.\, pr_2(h(x)))$$

is an equivalence.

proof. Define the inverse map

$$\text{choice}^{-1} : \sum \prod C(x, f(x)) \to \prod \sum C(x,y)$$

$$f : \prod_{x:A} B(x)^{x:A} \qquad\qquad x:A \quad y:B(x)$$

by $$\mathrm{choice}^{-1}(f,g) :\equiv \lambda x.(f(x), g(x)).$$

For the first homotopy it suffices to give an identification

$$\mathrm{choice}(\mathrm{choice}^{-1}(f,g)) = (f,g)$$

The left hand side computes to

$$\mathrm{choice}(\mathrm{choice}^{-1}(f,g)) \doteq \mathrm{choice}(\lambda x.(f(x),g(x))) \doteq (\lambda x.f(x), \lambda x.g(x))$$

By the η rule for function types $\doteq (f,g)$

For the second homotopy we require an identification

$$\mathrm{Choice}^{-1}(\mathrm{choice}(h)) = h$$

The left hand side computes to

$$\mathrm{choice}^{-1}(\mathrm{choice}\ h) \doteq \mathrm{choice}^{-1}(\lambda x.\,pr_1 h(x), \lambda x.\,pr_2 h(x)) \doteq \lambda x.(pr_1 h(x), pr_2 h(x))$$

We do not have a definitional equality relating $h(x) : \sum_{y:B(x)} C(x,y)$ and $(pr_1 h(x), pr_2 h(x))$. However we do have an identification between them b/c of our characterization of identity types of Σ types:

$$p,q : \sum_{x:D} E(x) \qquad (p = q) \simeq \sum_{\alpha : pr_1 p =_D pr_1 q} tr_D\, \alpha\, pr_2 p =_{D(pr_1 q)} pr_2 q$$

desired thing $\longleftarrow$ (refl, refl)

This gives us

$$\lambda x.\,\text{eq-pair}(\mathrm{refl}, \mathrm{refl}) : \mathrm{choice}^{-1}(\mathrm{choice}(h)) \sim h$$

Fun ext makes this an identification and then gives a htpy $\mathrm{choice}^{-1} \circ \mathrm{choice} \sim id$. ∎

§ Universal properties

Theorem (universal property of Σ-types)

Let B be a type family over A and let C be a type family over $\sum_{x:A} B(x)$. Then the map

$$\left(\prod_{z:\sum_{x:A}B(x)} C(z)\right) \xrightarrow{\text{ev-pair}} \left(\prod_{x:A}\prod_{y:B(x)} C(x,y)\right) \quad \text{is an equivalence}$$

$$f \longmapsto \lambda x\, \lambda y.\, f(x,y)$$

Proof: The inverse map is

$$\mathrm{ind}_\Sigma : \left(\prod_{x:A}\prod_{y:B(x)} C(x,y)\right) \to \left(\prod_{z:\sum_{x:A}B(x)} C(z)\right)$$

By the computation rules for Σ and Π types we have a homotopy

$$\text{refl-htpy} : \text{ev-pair} \circ \mathrm{ind}_\Sigma \sim \mathrm{id}.$$

Function extensionality provides the other homotopy $\mathrm{ind}_\Sigma \circ \text{ev-pair} \sim \mathrm{id}$

This requires an identification for all $f : \prod_{z:\sum_{x:A}B(x)} C(z)$

$$\mathrm{ind}_\Sigma(\lambda x\, \lambda y.\, f(x,y)) = f$$

By fun ext

$$\prod_{z:\sum_{x:A}B(x)} \mathrm{ind}_\Sigma(\lambda x \lambda y. f(x,y))(z) = f(z).$$

By Σ-ind we can reduce to the case ~~where~~ of a pair (a,b).

Both sides are definitionally $f(a,b)$. $\blacksquare$

Cor $(A \times B \to C) \simeq (A \to B \to C)$

§ Universal property of identity types

The based identity type for A and $a:A$ is a family

$\lambda x.\, a =_A x : A \to \mathcal{U}$.

Consider $B : A \to \mathcal{U}$. B maps $x:A$ to a type $B(x):\mathcal{U}$.

B also maps a path $p : x = y$, $\mathrm{tr}\, B\, p : B(x) \to B(y)$.

A "natural transformation" from the identity type family into B is a term of type

$$\prod_{x:A} (a =_A x) \to B(x)$$

Theorem (Yoneda lemma) For any $A:\mathcal{U}$, $a:A$, $B : A \to \mathcal{U}$, $C : A \to \mathcal{U}$. The following map is an equivalence

$$\left(\prod_{x:A} (a =_A x) \to B(x) \right) \xrightarrow{\mathrm{ev}\text{-}\mathrm{refl}(a)} B(a) .$$

$\prod_{x:A} B(x) \to C(x)$.

$\alpha : a' =_A a$

The full universal property of identity types is stronger.

Theorem (dependent Yoneda lemma/universal property of identity types).

Let A be a type, $a:A$, $B(x,p)$ a family over $x:A$, $p:a=x$.

Then

$$\mathrm{ev}\text{-}\mathrm{refl} : \left(\prod_{x:A} \prod_{p:a=x} B(x,p) \right) \longrightarrow B(a, \mathrm{refl}_a)$$

is an equivalence.

proof: The inverse map is

$$\text{path-ind}_a : B(a, \text{refl}_a) \to \prod_{x:A} \prod_{p:a=x} B(x,p)$$

This is a section by the computation rule for id types.

For the other homotopy $\text{path-ind}_a \circ \text{ev-refl} \sim \text{id}$

let $f : \prod_{x:A} \prod_{p:a=x} B(x,p)$ we require

$$\text{path-ind}_a(f(a, \text{refl}_a)) = f$$

By <u>fun ext</u> it suffices to show

$$\prod_{x:A} \prod_{p:a=x} (\text{path-ind}_a(f(a,\text{refl}_a))(x,p) = f(x,p))$$

By path induction it suffices to show

$$(\text{path-ind}_a(f(a,\text{refl}_a))(a, \text{refl}_a) = f(a, \text{refl}_a))$$

This holds definitionally by computation rules. $\square$