1 Co-product Paths

As in the HoTT book, we wish to prove the following characterization of paths in a coproduct type $A + B$:

1. $\mathsf{inl}(M) =_{A+B} \mathsf{inl}(M') \simeq M =_A M'$;
2. $\mathsf{inr}(N) =_{A+B} \mathsf{inr}(N') \simeq N =_B N'$;
3. $\mathsf{inl}(\_) =_{A+B} \mathsf{inr}(\_) \simeq \mathsf{void}$;
4. $\mathsf{inr}(\_) =_{A+B} \mathsf{inl}(\_) \simeq \mathsf{void}$.

Define the family

$$u : A + B, v : A + B \vdash F[u, v] : \mathcal{U}$$

so that the following definitional equivalences hold:

1. $F[\mathsf{inl}(M), \mathsf{inl}(M')] \equiv M =_A M'$;
2. $F[\mathsf{inr}(M), \mathsf{inr}(M')] \equiv M =_A M'$;
3. $F[\mathsf{inl}(\_), \mathsf{inr}(\_)] \equiv \mathsf{void}$;
4. $F[\mathsf{inr}(\_), \mathsf{inl}(\_)] \equiv \mathsf{void}$.

This is easily achieved by a nested case analysis on u and v, respectively, with motive $\mathcal{U}$ in each case. The correspondence between the defnition of F and the desired theorem is evident. The point is that F cancels the injections definitionally, so that no particular mention need be made of this in the proof.

Lemma 1. *There is a term L of type $\prod_{z:A+B} F[z, z]$ such that $L(\mathsf{inl}(x)) \equiv \mathsf{refl}_A(x)$ and $L(\mathsf{inr}(y)) \equiv \mathsf{refl}_B(y)$.*

Proof. Construct L by abstracting over z, then performing a case analysis on z:

1. $x : A \vdash \mathsf{refl}_A(x) : F[\mathsf{inl}(x), \mathsf{inl}(x)]$.
2. $y : B \vdash \mathsf{refl}_A(y) : F[\mathsf{inr}(y), \mathsf{inr}(y)]$.

□

We wish to show that

$$\prod_{z,z':A+B} z =_{A+B} z' \simeq F[z, z'],$$

from which the desired result follows immediately by definition of F.

First we exhibit the function

$$f : \prod_{z:A+B} \prod_{z':A+B} z =_{A+B} z' \to F[z, z']$$

given as follows:

$$\lambda z.\lambda z'.\lambda p.\mathsf{J}[u.v._.F[u,v]](p; x.L(x)).$$

Lemma 1 does all the work: if $M : A + B$, then

$$f(M)(M)(\mathsf{refl}_{A+B}(M)) \equiv L(M).$$

We then exhibit a quasi-inverse for f,

$$g : \prod_{z:A+B} \prod_{z':A+B} F[z,z'] \to z =_{A+B} z'$$

given by a nested case analysis on z and z' based on the following data:

1. $x : A, x' : A, z : F[\mathsf{inl}(x), \mathsf{inl}(x')] \vdash \mathsf{ap}_{\mathsf{inl}(-)}(z) : \mathsf{inl}(x) =_{A+B} \mathsf{inl}(x')$;
2. $x : A, y' : B, z : F[\mathsf{inl}(x), \mathsf{inr}(y')] \vdash \mathsf{abort}(z) : \mathsf{inl}(x) =_{A+B} \mathsf{inr}(y')$.
3. $y : B, y' : B, z : F[\mathsf{inr}(x), \mathsf{inr}(x')] \vdash \mathsf{ap}_{\mathsf{inr}(-)}(z) : \mathsf{inr}(x) =_{A+B} \mathsf{inr}(x')$;
4. $x : A, y' : B, z : F[\mathsf{inr}(x), \mathsf{inl}(y')] \vdash \mathsf{abort}(z) : \mathsf{inr}(x) =_{A+B} \mathsf{inl}(y')$.

Here we are relying on the definitional properties of the family F to justify the given typings. Notice that the λ-abstraction of the third argument to g must occur *inside* the case analysis in order to propagate the correct branch information (*cf.* the hacky treatment of this issue in Haskell's so-called GADT's.)

Lemma 2. *The following types are inhabited:*

1. $x : A \vdash \_ : g(\mathsf{inl}(x))(\mathsf{inl}(x))(\mathsf{refl}_A(x)) =_{A+B} \mathsf{refl}_{A+B}(\mathsf{inl}(x))$;
2. $y : B \vdash \_ : g(\mathsf{inr}(x))(\mathsf{inr}(x))(\mathsf{refl}_B(y)) =_{A+B} \mathsf{refl}_{A+B}(\mathsf{inr}(y))$.

To complete the proof we need only exhibit witnesses to the fact that for $z, z' : A + B$, the function $g' = g(z)(z')$ is right and left inverse to the function $f' = f(z)(z')$, up to higher homotopy.

1. $z : A + B, z' : A + B, w : F(z, z') \vdash \alpha : f'(g'(w)) =_{F[z,z']} w$.
2. $z : A + B, z' : A + B, w : z =_{A+B} z' \vdash \beta : g'(f'(w)) =_{z=_{A+B}z'} w$.

The first is proved by a nested case analysis on z and z', using the definitional equivalences governing F, either aborting, or performing a path induction on w, appealing to Lemma 2 to complete the proof. The second is proved by induction on w, using a case analysis on the generic $u : A + B$ so that we may appeal to Lemma 2 to complete the proof.

2 Application to Paths

The idea is to adapt the above proof for coproducts, except using propositional equivalences for the family F obtained from the assumption that f has a quasi-inverse.

We are given $f : A \to B$ and the following data showing that f has a quasi-inverse:

1. $f^{-1} : B \to A$;
2. $\alpha : \prod_{a:A} f^{-1}(f(a)) =_A a$;
3. $\beta : \prod_{b:B} f(f^{-1}(b)) =_B b$.

We are to show that ap_f has a quasi-inverse, which amounts to giving the following data:

1. ap_f^{-1}, which is taken to be $\lambda q.\alpha(a)^{-1} \cdot \mathsf{ap}_{f^{-1}}(q) \cdot \alpha(a')$;
2. $\alpha' : \prod_{a:A} \prod_{a':A} \prod_{p:a=_A a'} \mathsf{ap}_f^{-1}(\mathsf{ap}_f(p)) =_{a=_A a'} p$;
3. $\beta' : \prod_{a:A} \prod_{a':A} \prod_{q:f(a)=_B f(a')} \mathsf{ap}_f(\mathsf{ap}_f^{-1}(q)) =_{f(a)=_B f(a')} q$.

The construction of α' takes the form of a path induction on p, reducing the problem to the case of reflexivity for a generic x of type A. This presents no difficulties, because the end points of the path in question are variables that also occur in the motive.

The construction of β' is more difficult, because the evident source of path induction, q, has as end points $f(a)$ and $f(a')$, which will appear in the conclusion of the proof. More precisely, if F is the motive for a path induction on q, then the conclusions will be of the form $F[f(a), f(a'), q]$, whereas the desired conclusion involves just a, a', and q.

We must choose the path on which to induct and the motive for the induction in such a way that the desired conclusion follows from the corresponding instance of the motive. One approach is to induct on $\mathsf{ap}_{f^{-1}}(q)$, which has type

$$f^{-1}(f(a)) =_A f^{-1}(f(a')).$$

Using the quasi-inverse for f this type may be shown to be propositionally equal to the type $a =_A a'$, so any element of the former type may be transported to the latter, and *vice versa*.

An appropriate motive for the induction is the type family $u{:}A, v{:}A, w{:}u =_A v \vdash F : \mathcal{U}$ defined by the equality type

$$\mathsf{ap}_f(\alpha(u)^{-1} \cdot \mathsf{ap}_{f^{-1}}(\mathsf{ap}_f(w)) \cdot \alpha(v)) =_{f(u)=_B f(v)} \mathsf{ap}_f w.$$

With F as motive the path induction on $\mathsf{ap}_{f^{-1}}(q)$ yields an inhabitant of the type

$$F[f^{-1}(f(a)), f^{-1}(f(a')), \mathsf{ap}_{f^{-1}}(q)],$$

which is equal to

$$\mathsf{ap}_f(\alpha(f^{-1}(f(a)))^{-1} \cdot \mathsf{ap}_{f^{-1}}(\mathsf{ap}_f(\mathsf{ap}_{f^{-1}}(q))) \cdot \alpha(f^{-1}(f(a')))) = \mathsf{ap}_f(\mathsf{ap}_{f^{-1}}(q)).$$

Using the quasi-inverse for f, we may show that this equation is equal to the equation

$$\mathsf{ap}_f(\alpha(a)^{-1} \cdot \mathsf{ap}_{f^{-1}}(q) \cdot \alpha(a')) = q,$$

which is the desired conclusion.

It remains to show, then, that

$$x{:}A \vdash \_ : F[x, x, \mathsf{refl}_A(x)],$$

which is to say that

$$x{:}A \vdash \_ : \mathsf{ap}_f(\alpha(x)^{-1} \cdot \mathsf{ap}_{f^{-1}}(\mathsf{ap}_f(\mathsf{refl}_A(x))) \cdot \alpha(x)) = \mathsf{ap}_f(\mathsf{refl}_A(x)).$$

Now $\mathsf{ap}_f(\mathsf{refl}_A(x)) \equiv \mathsf{refl}_B(f(x))$, and $\mathsf{ap}_f^{-1}(\mathsf{refl}_B(f(x)))$, so this amounts to showing

$$x{:}A \vdash \_ : \mathsf{ap}_f(\alpha(x)^{-1} \cdot \mathsf{refl}_B(f^{-1}(f(x))) \cdot \alpha(x)) = \mathsf{refl}_B(f(x)).$$

Using the unit laws for path concatenation this reduces to showing

$$y : B \vdash \_ : \mathsf{ap}_f(\alpha(y)^{-1} \cdot \alpha(y)) =_B \mathsf{refl}_B(y).$$

Applying the inverse law for paths, this is just

$$y : B \vdash \_ : \mathsf{ap}_f(\mathsf{refl}_B(f^{-1}(y))) =_B \mathsf{refl}_B(y).$$

Finally, $\mathsf{ap}_f(\mathsf{refl}_B(f^{-1}(y))) \equiv \mathsf{refl}_B(f(f^{-1}(y)))$, and we have

$$y : B \vdash \_ : \mathsf{refl}_B(f(f^{-1}(y))) =_B \mathsf{refl}_B(y),$$

using the quasi-inverse of f, which completes the argument.

Throughout I am tacitly using the principle that

$$\mathsf{trans}[x.x](p) : A \to A'$$

whenever $p : A =_{\mathcal{U}} A'$, which is to say that one may transport objects of a type A to an object of an equal type A' in the same universe. This move corresponds to the implicit uses of definitional equality of types in the characterization of the paths in a coproduct type.