Categories for the Working Homotopy Type Theorist

Everything has to come to an end. Even chapter two.

by

Merlin Göttlinger

July 26, 2017

Structure

Motivation

We have seen how to work with

- Σ-types
- (cartesian) product-types
- Π-types
- coproduct-types

Now we are going to see how to derive the correct notion of equality for structures built out of those and how our intuition from category theory also serves us well here.

Semigroups

Definition (Semigroup)

Given a type A, the type $\mathit{SemigroupStr}(A)$ of semigroup structures with carrier A is defined by

$$\mathit{SemigroupStr}(A) :\equiv \sum_{(m:A\to A\to A)} \prod_{(x,y,z:A)} m(x, m(y,z)) = m(m(x,y),z)$$

A semigroup is a type together with such a structure:

$$\mathit{Semigroup} :\equiv \sum_{(A:\mathcal{U})} \mathit{SemigroupStr}(A)$$

Univalence to the Rescue

Lifting Equivalences

From classical mathematics we are used to bijections on sets being isomorphisms. If we have an iso between A and B we therefore also should have an iso between $SemigroupStr(A)$ and $SemigroupStr(B)$.
In our brave new world given an $e : A \simeq B$ we should have:

$$transport^{SemigroupStr}(ua(e)) : SemigroupStr(A) \to SemigroupStr(B)$$

which furthermore should be an equivalence to reflect what we knew previously.

Lifting Equivalences

Given that

$$ua : (A \simeq B) \to (A =_{\mathcal{U}} B)$$

produces a path which we can invert and $transport^C(\alpha)$ is always an equivalence with inverse $transport^C(\alpha^{-1})$ this is indeed an equivalence.
But how does it operate exactly?

How does it work?

Given that *ua* is an axiom we don't actually know how it does what it does.
We do however know a few things about *transport* which we can use to figure out what is going on.
Let $(m, a) : SemigroupStr(A)$
How does the induced semigroup structure on B given by

$$transport^{SemigroupStr}(ua(e), (m, a))$$

work?

How does it work?

First, because $\mathit{SemigroupStr}(X)$ is defined to be a Σ-type,

$$\mathit{transport}^{\mathit{SemigroupStr}}(\mathit{ua}(e), (m, a)) = (m', a')$$

where m' is an induced multiplication operation on B

$$m' : B \to B \to B$$
$$m'(b_1, b_2) :\equiv \mathit{transport}^{X \mapsto (X \to X \to X)}(\mathit{ua}(e), m)(b_1, b_2)$$

and a' an induced proof that m' is associative.

How does it work?

What does a' look like?
Because *SemigroupStr* is a Σ-type:

$$\begin{aligned} a' &: \mathrm{Assoc}(B, m') \\ a' &:\equiv \mathit{transport}^{(X,m)\mapsto \mathrm{Assoc}(X,m)}((\mathit{pair}^{=}(\mathit{ua}(e), \mathit{refl}_{m'})), a) \end{aligned} \tag{1}$$

where $\mathrm{Assoc}(X, m)$ is the type

$$\prod_{(x,y,z:X)} m(x, m(y, z)) = m(m(x, y), z)$$

How does it work?

By function extensionality, it suffices to investigate the behavior of m' when applied to arguments $b_1, b_2 : B$.

Recall: Transport Arrow

$$transport^{A \to B}(p, f) = x \mapsto transport^{B}(p, f(transport^{A}(p^{-1}, x)))$$

By applying this, we have that $m'(b_1, b_2)$ is equal to

$$\begin{aligned}
& tr^{X \mapsto (X \to X \to X)}(ua(e), m)(b_1, b_2) \\
= \; & tr^{X \mapsto (X \to X)}(ua(e), y \mapsto m(tr^{X \mapsto X}(ua(e)^{-1}, b_1), y))(b_2) \\
= \; & tr^{X \mapsto X}(ua(e), m(tr^{X \mapsto X}(ua(e)^{-1}, b_1), tr^{X \mapsto X}(ua(e)^{-1}, b_2)))
\end{aligned}$$

where *tr* means *transport* to fit on slide.

How does it work?

Then, because *ua* is quasi-inverse to $\mathit{transport}^{X \mapsto X}$, this is equal to

$$e(m(e^{-1}(b_1), e^{-1}(b_2))).$$

Thus, given two elements of B, the induced multiplication m' sends them to A using the equivalence e, multiplies them in A, and then brings the result back to B by e, just as we would expect from our previous knowledge about isos.

What about the Associativity?

We could try to find out how the proof is translated but it is easier to quickly proove that m' is associative:

$$
\begin{aligned}
m'(m'(b_1, b_2), b_3) &= e(m(e^{-1}(m'(b_1, b_2)), e^{-1}(b_3))) \\
&= e(m(e^{-1}(e(m(e^{-1}(b_1), e^{-1}(b_2)))), e^{-1}(b_3))) \\
&= e(m(m(e^{-1}(b_1), e^{-1}(b_2)), e^{-1}(b_3))) \\
&= e(m(e^{-1}(b_1), m(e^{-1}(b_2), e^{-1}(b_3)))) \\
&= e(m(e^{-1}(b_1), e^{-1}(e(m(e^{-1}(b_2), e^{-1}(b_3)))))) \\
&= e(m(e^{-1}(b_1), e^{-1}(m'(b_2, b_3)))) \\
&= m'(b_1, m'(b_2, b_3)).
\end{aligned} \tag{2}
$$

Equality of Semigroups

Now that we somewhat know how transporting of semigroups works, what should their identiy-type look like? Since semigroups are defined as Σ-types

$$(A, m, a) =_{Semigroup} (B, m', a')$$

is equal to the type of pairs

$$p_1 : A =_{\mathcal{U}} B \qquad \text{and}$$
$$p_2 : transport^{SemigroupStr}(p_1, (m, a)) = (m', a').$$

Equality of Semigroups

From univalence we get that p_1 is $ua(e)$ from some equivalence e.

By $(w = w') \simeq \sum_{(w,w' : \sum_{(x:A)} P(x))} p_*(pr_2(w)) = pr_1(w')$, function extensionality, and the above analysis of $transport^{SemigroupStr}$, we get that p_2 is equivalent to a pair of proofs:

- $\prod_{(y_1,y_2:B)} e(m(e^{-1}(y_1), e^{-1}(y_2))) = m'(y_1, y_2)$
- a' is equal to the induced associativity proof constructed from a when transporting

Equality of Semigroups

But by cancellation of inverses the first one is equivalent to

$$\prod_{(x_1, x_2 : A)} e(m(x_1, x_2)) = m'(e(x_1), e(x_2)).$$

This says that e commutes with the binary operation, in the sense that it takes multiplication in A (i.e. m) to multiplication in B (i.e. m').

A similar rearrangement is possible for the equation relating a and a'.

Thus, an equality of semigroups consists exactly of an equivalence on the carrier types that commutes with the semigroup structure.

Abstracting

Thus, we have arrived at a standard definition of a *semigroup isomorphism*: a bijection on the carrier sets that preserves the multiplication operation. It is also possible to use the category-theoretic definition of isomorphism, by defining a *semigroup homomorphism* to be a map that preserves the multiplication, and arrive at the conclusion that equality of semigroups is the same as two mutually inverse homomorphisms.

So, thanks to univalence, semigroups are equal precisely when they are isomorphic as algebraic structures. In homotopy type theory, all constructions of mathematical structures automatically respect isomorphisms, without any tedious proofs or abuse of notation.

A Hint of Category Theory

Universal Property of Products

We have seen that concepts from category theory that we know and love are derivable in HoTT as well.
So what about universal properties?
E.g. the one of products:

$$u : (X \to A \times B) \to (X \to A) \times (X \to B) \tag{3}$$

defined by $f \mapsto (pr_1 \circ f, pr_2 \circ f)$.

Universal Property of Products

Theorem (Universal Prop Equiv)

The universal property of products is an equivalence.

Proof direction one.

Let j be the function $(g, h) \mapsto (x \mapsto (g(x), h(x)))$.
Now given $f : X \to A \times B$ we have

$$(j \circ u)(f) \equiv x \mapsto (pr_1(f(x)), pr_2(f(x))) \tag{4}$$

By $(x =_{A\times B} y \simeq (pr_1(x) = pr_1(y)) \times (pr_2(x) = pr_2(y))$, for any $x : X$ we have

$$(pr_1(f(x)), pr_2(f(x))) = f(x)$$

Thus, by function extensionality, $(j \circ u)(f)$ is equal to f. □

Universal Property of Products

Proof direction two.
On the other hand, given $(g : X \to A, h : X \to B)$, we have

$$(u \circ j)(g, h) \equiv (x \mapsto g(x), x \mapsto h(x))$$

By the uniqueness principle for functions, this is (judgmentally) equal to (g, h). $\square$

Beyond Category Theory

What about Π-types?

In fact, we also have a dependently typed version of this universal property. Suppose given a type X and type families $A, B : X \to \mathcal{U}$. Then we have a function

$$\prod_{(x:X)} (A(x) \times B(x)) \to \prod_{(x:X)} A(x) \times \prod_{(x:X)} B(x) \qquad (5)$$

defined as before by $f \mapsto (pr_1 \circ f, pr_2 \circ f)$.

Theorem (Universal Prop Dep Equiv)

The universal property of products of Π-types is an equivalence.

What about Σ-types?

Just as Σ-types are a generalization of cartesian products, they satisfy a generalized version of this universal property. Let X be a type, $A : X \to \mathcal{U}$ be a type family and $P : \prod_{(x:X)} A(x) \to \mathcal{U}$. Then we have a function

$$u : \left(\prod_{(x:X)} \sum_{(a:A(x))} P(x, a) \right) \to \sum_{(g:\prod_{(x:X)} A(x))} \prod_{(x:X)} P(x, g(x)) \qquad (6)$$

Note that if we have $P(x, a) :\equiv B(x)$ for some $B : X \to \mathcal{U}$, then this reduces to the equivalence we have seen before.

What about Σ-types?

Theorem (Universal Prop Double Dep Equiv)

The universal property of Σ-types of Π-types is an equivalence.

What about Σ-types?

Proof direction one.

Let j be $(g : \prod_{(x:X)} A(x), h : \prod_{(x:X)} P(x, g(x))) \mapsto (x \mapsto (g(x), h(x)))$.

Now given $f : \prod_{(x:X)} \sum_{(a:A(x))} P(x, a)$, we have

$$(j \circ u)(f) \equiv x \mapsto (pr_1(f(x)), pr_2(f(x))) \tag{7}$$

Now for any $x : X$, by the uniqueness principle for Σ-types we have

$$(pr_1(f(x)), pr_2(f(x))) = f(x).$$

Thus, by function extensionality, $(j \circ u)(f)$ is equal to f. $\square$

What about Σ-types?

Proof direction two.

On the other hand, given $(g : \prod_{(x:X)} A(x), h : \prod_{(x:X)} P(x, g(x)))$, we have

$$(u \circ j)(g, h) \equiv (x \mapsto g(x), x \mapsto h(x))$$

which is judgmentally equal to (g, h) as before. $\square$

An old friend?

If you look closely…

$$\left(\prod_{(x:X)} \sum_{(a:A(x))} P(x,a)\right) \to \sum_{(g:\prod_{(x:X)} A(x))} \prod_{(x:X)} P(x, g(x))$$

looks familiar.

- "for all $x : X$ there exists an $a : A(x)$ such that $P(x, a)$", and
- "there exists a choice function $g : \prod_{(x:X)} A(x)$ such that for all $x : X$ we have $P(x, g(x))$".

Axiom of Choice

Thus, this says that not only is the axiom of choice “true” but also that its antecedent is actually equivalent to its conclusion.

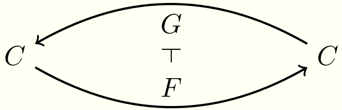
Other Old Friends From CT

Adjunctions

Let $F, G : C \to C$ be endofunctors on some category C and $A, B \in Obj(C)$ such that

$$C(FA, B) \simeq C(A, GB)$$

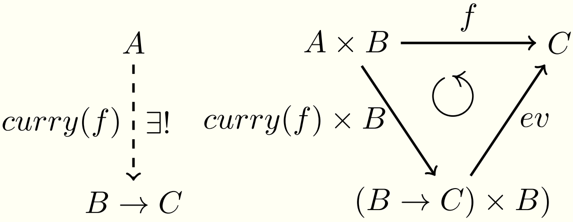
Then $F \dashv G$. "F is left adjoint to G"



Adjunctions

The above universal property for pairs is for “mapping in”.
However, pair types also have a universal property for “mapping out”:
The cartesian closure adjunction:

$$((A \times B) \rightarrow C) \simeq (A \rightarrow (B \rightarrow C))$$



Adjunctions with $\prod$-types

The dependent version of this is formulated for a type family $C : A \times B \to \mathcal{U}$:

$$\prod_{(w:A\times B)} C(w) \simeq \prod_{(x:A)} \prod_{(y:B)} C(x,y)$$

Here the right-to-left function is simply the induction principle for $A \times B$, while the left-to-right is evaluation at a pair.

Adjunctions with Σ-types

There is also a version for Σ-types:

$$\prod_{(w:\sum_{(x:A)} B(x))} C(w) \simeq \prod_{(x:A)} \prod_{(y:B(x))} C(x,y). \tag{8}$$

Again, the right-to-left function is the induction principle.

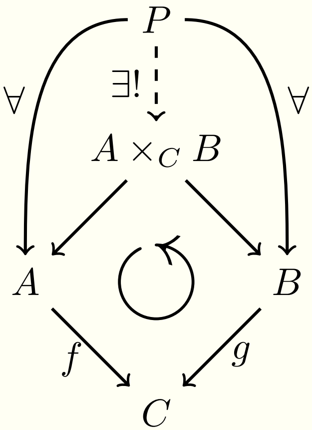
Path Induction

Some other induction principles are also part of universal properties of this sort. For instance, path induction is the right-to-left direction of an equivalence as follows:

$$\prod_{(x:A)} \prod_{(p:a=x)} B(x,p) \simeq B(a, refl_a) \tag{9}$$

for any $a : A$ and type family $B : \prod_{(x:A)} (a = x) \to \mathcal{U}$.

Pullbacks

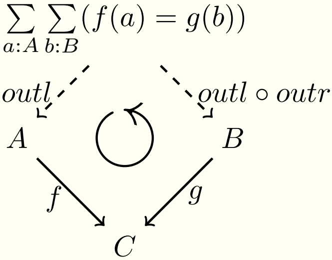


A pullback $A \times_C B$ is an object with two projections to A and B such that the commutes and every other competitor P can be mapped uniquely to it.

Pullbacks

For pullbacks, the expected explicit construction works: given $f : A \to C$ and $g : B \to C$, we define

$$A \times_C B :\equiv \sum_{(a:A)} \sum_{(b:B)} (f(a) = g(b)) \tag{10}$$



Some more general homotopy limits can be constructed in a similar way, but for colimits we will need a new ingredient.

Extensional vs Intensional TT

Our identity types are *intensional* because they distiguish objects based on their definition even if they have the same *observable behaviour*.
We make our type theory a bit more *extensional* by including function extensionality and univalence.
A fully extensional type theory would also have

$$\mathit{reflection}(p : x = y) :\equiv x \equiv y$$

which forces the two notions of equality to coincide.
You would gain from this that function extensionality would be derivable but at the cost of disallowing univalence.

Exercises

Coproducts UP

Prove that coproducts have the expected universal property

$$(A + B \to X) \simeq (A \to X) \times (B \to X)$$

Let

$$j :\equiv (abx : A + B \to X) \mapsto (abx \circ inl, abx \circ inr)$$

and

$$i :\equiv (ax : A \to X, bx : B \to X) \mapsto (ab \mapsto rec_{A+B}(X, ax, bx, ab))$$

Those are inverses.

Coproducts UP

Proof direction one.

By induction on cartesian products it suffices to prove the case of a pair ax, bx

$$
\begin{aligned}
(j \circ i)(ax, bx) \equiv \\
&\equiv (a \mapsto rec_{A+B}(X, ax, bx, inl(a)), \\
&\quad\; b \mapsto rec_{A+B}(X, ax, bx, inr(b))) \\
&\equiv (a \mapsto ax(a), b \mapsto bx(a))
\end{aligned}
$$

By uniqueness principle of functions this is equal to (ax, bx). □

Coproducts UP

Proof direction two.
Composing the two functions we get

$$(i \circ j)(abx) \equiv ab \mapsto rec_{A+B}(X, abx \circ inl, abx \circ inr, ab)$$

We then show by induction over coproducts that given some f and X

$$rec_{A+B}(X, f \circ inl, f \circ inr) = f$$

Thus

$$(i \circ j)(abx) = ab \mapsto abx(ab)$$

By uniqueness principle of functions this is equal to abx $\square$

Coproducts UP

Can you generalize this to an equivalence involving dependent functions?

$$\left(\prod_{(ab:A+B)} X(ab)\right) \simeq \left(\prod_{(a:A)} X(inl(a))\right) \times \left(\prod_{(b:B)} X(inr(b))\right)$$

Let

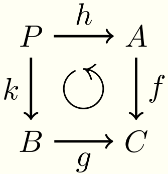
$$i :\equiv abx \mapsto (a \mapsto abx(inl(a)), b \mapsto abx(inr(b)))$$

and

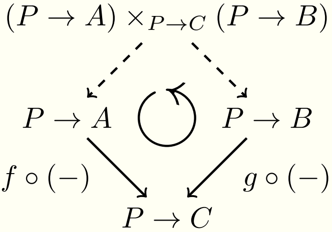
$$j :\equiv (ax, bx) \mapsto (ab \mapsto rec_{A+B}(X(ab), ax, bx, ab))$$

Definition (Commutative Square)

A (homotopy) commutative square



Note that this is exactly an element of the pullback $(P \to A) \times_{P \to C} (P \to B)$ as defined in (10).



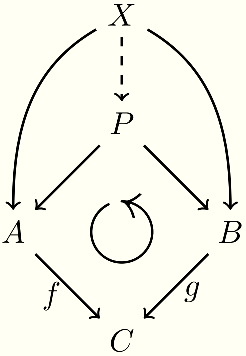
Pullbacks

Definition (Pullback Square)

A commutative square is called a (homotopy) pullback square if for any X, the induced map

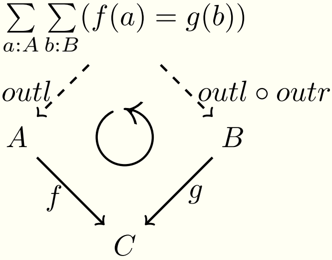
$$(X \to P) \to (X \to A) \times_{(X \to C)} (X \to B)$$

is an equivalence.



Pullbacks

Prove that the pullback $P :\equiv A \times_C B$ defined in (10) is the corner of a pullback square.



Pullbacks

$$(X \to A \times_C B) \simeq (X \to A) \times_{X \to C} (X \to B)$$

Let

$$i :\equiv xp \mapsto (outl \circ xp, (outl \circ outr \circ xp, oprf))$$

where

$$oprf :\equiv funext(x \mapsto (outr \circ outr \circ xp)(x))$$

$$j :\equiv (xa, (xb, prf : f{\circ}xa = g{\circ}xb)) \mapsto (x \mapsto (xa(x), (xb(x), nprf)))$$

where

$$nprf :\equiv happly(prf)(x) : f(xa(x)) = g(xb(x)))))$$

Pullback pasting

Suppose given two commutative squares

$$\begin{array}{ccccc} A & \longrightarrow & C & \longrightarrow & E \\ \downarrow & & \downarrow & & \downarrow \\ B & \longrightarrow & D & \longrightarrow & F \end{array}$$

and suppose that the right-hand square is a pullback square. Prove that the left-hand square is a pullback square if and only if the outer rectangle is a pullback square.