HOTT SEMINAR LECTURE NOTES - COVERING 4.1 - 4.5 (AND A BIT OF 4.8) FROM THE BOOK

ALEXANDER YOM DIN

Contents

0. Recollections 1: Basics

0.1. **Types and terms.** The basic axiomatic object in type theory is a *type*. A type has *terms* (or *inhabitants*). We write $x : X$ to denote that x is a term of the type X.

By a statement (resp. supposition) of the form $a : X$ we mean that we will construct (resp. suppose already constructed) a term of the type X and we name that term a. By a statement of the form X, we mean the same, except that we don't want to introduce a notation for the constructed term (thus, the difference is just a matter of exposition, the real meaning is the same).

All what we do in the type-theoretical approach is to construct terms of types, when given previously constructed terms of types, by some rules. Thus, one contrast with set theory is that claims themselves are simply the announcements of terms to be constructed given terms already constructed, and proofs of claims are the constructions[1]. Using the function type, one can even contract this into saying that a claim is a type, and a proof of it is the inhabitation of that type (by inhabiting a type we mean constructing a term of that type).

It is convenient to have the type of types $Types$, whose terms are types, and we ignore here the obvious need of some hierarchy for such a notion to be acceptable.

0.2. **Identity types.** Given $X : Types$ and $x_1, x_2 : X$, one axiomatially introduces the *identity type* $x_1 = x_2$. As far as I understand, this is the main departure from set theory; Roughly speaking, one first (even historically, I think - ancient Greece and so on) sees the need in studying manipulations with truth values (True and False, or contractible and empty in the topological situtation). In a set, the inter-relations between (or identifications of) two inhabitants are described by a truth value. Then one has a 1-groupoid, where the inter-relations between two inhabitants form a set, and so on, leading to the notion of an ∞-groupoid. In a type, the inter-relations between two inhabitants form again a type - making types suitable for an definition of ∞-groupoids which will be "at once".

We think of a type X as a space/∞-groupoid in the sense of homotopy theory/higher category theory (and of terms of a type as the points/objects of the space/∞-groupoid respectively). Of $x_1 = x_2$ we think as the space of paths from x_1 to x_2 (in the space interpretation), or the ∞-groupoid of isomorphisms between x_1 and x_2 (in the ∞-groupoid interpretation).

0.3. **Function, product and sum types.** Given $X, Y : Types$, one axiomatically introduces the *function type* $X \to Y$. One has some standard rules, such as given $f : X \to Y$ and $x : X$, an $f(x) : Y$ is constructed.

Given $X : Types$ and $Y : X \to Types$ (an "*X-parametrized type*"), one axiomatically introduces the *product type* $\prod_{x:X} Y(x)$ and the *sum type* $\sum_{x:X} Y(x)$, we some standard rules.

In the space interpretation, the product $\prod_{x:X} Y(x)$ is the homotopy limit over the fundamental ∞-groupoid corresponding to the space X, of spaces $Y(x)$ (and similarly for sums and homotopy colimits).

One can define composition of functions with some standard rules, i.e.

$$composition : \prod_{X,Y,Z:Types} \prod_{f:X\to Y, g:Y\to Z} \Big(X \to Z\Big).$$

We denote

$$g \circ f \quad :\equiv \quad composition(X, Y, Z, f, g).$$

We also have the identity function $id_X : X \to X$, etc. All this is because to give a function $f : X \to Y$ is the same as to produce an element of Y each time an element of X is given, and then it is clear how to define composition and the identity function, and so on. Notice that associativity will be satisfied on the nose! In this sense, types are already conveniently an $(\infty, 1)$-category.

[1]Although maybe one can modify set theory a bit to similarly emphasize this.

0.4. **Path induction.** One introduces axiomatically, for all types $X : Types$ and terms $x : X$, the "*the path witnessing reflexivity*" $refl_X : x = x$ (thought of as the constant path in the space interpretation and the identity automorphism in the ∞-groupoid interpretation). What we really mean by this is that one axiomatically inhabits $\prod_{X:Types} \prod_{x:X} (x = x)$. This is an indication to the line of thought, by which all constructions in type theory are always "functorial".

A very important principle for identity types is *path induction*:

Axiom 0.1 (Path induction)**.** Let

$$X : Types, \quad Y : \prod_{x_1, x_2 : X} \prod_{\alpha : x_1 = x_2} Types.$$

Then we have

$$pathind_{X,Y} : \prod_{x:X} Y(x, x, refl_x) \to Y$$

such that $pathind(f)(x, x, refl_x) :\equiv f(x)$[2]. In words: If for all $x_1, x_2 : X$ and $\alpha : x_1 = x_2$ we are given a type $Y(x_1, x_2, \alpha)$, and if we constructed terms of $Y(x, x, refl_x)$ for all $x : X$, then also constructed are terms of $Y(x_1, x_2, \alpha)$ for all $x_1, x_2 : X$ and $\alpha : x_1 = x_2$.

In the space interpretation, path induction reflects the homotopy equivalence $X \to X \underset{X}{\times} X$ (where the fiber product is the homotopy fiber product).

By using path induction one inhabits, for all types $X, Y : Types$ and function $f : X \to Y$, the type $\prod_{x_1, x_2 : X} \Big((x_1 = x_2) \to (f(x_1) = f(x_2)) \Big)$. Intuitively, functions between types are always functorial.

By using path induction, one can define *concatenation of paths*, i.e.

$$concat_X : \prod_{x_1, x_2, x_3 : X} \prod_{\alpha : x_1 = x_2, \beta : x_2 = x_3} \Big(x_1 = x_3 \Big).$$

We then denote, given $\alpha : x_1 = x_2, \beta : x_2 = x_3$,

$$\beta \bullet \alpha \quad :\equiv \quad concat_X(x_1, x_2, x_3, \alpha, \beta) : x_1 = x_3.$$

One can similarly define the *inverse path*:

$$inv_X : \prod_{x_1, x_2 : X} \prod_{\alpha : x_1 = x_2} \Big(x_2 = x_1 \Big)$$

(we denote the inverse of $\alpha : x_1 = x_2$ by $\alpha^{-1} : x_2 = x_1$), and verify the various identities one expects, such as associativity of concatenation and so on (in the homotopical sense).

[2]The symbol $:\equiv$ means that the left hand side is equal to the right hand side by a meta-theoretical decree, i.e. one has a complete identification, part of the rules of the game, not constituting an inhabitation of a type but rather a complete interchangeability on the level of what strings can be written - so not related to the internal notion of equality types using the "=" symbol.

0.5. **Pointwise identities.** Let

$$Y : X \to Types, \quad f, g : \prod_{x:X} Y(x).$$

The *type of pointwise identities (or homotopies)* is

$$f \overset{pw}{=} g \quad :\equiv \quad \prod_{x:X} \Big(f(x) = g(x) \Big).$$

One defines $refl_f^{pw}$, concatenation, inverses, application of a function to a pointwise identity, and so on - in an evident way.

Let us reiterate the meaning of some notation which we will use. Given $f : X \to Y$ and $g, h : Z \to X$ and $\alpha : g \overset{pw}{=} h$, we will denote by $f(\alpha) : f \circ g \overset{pw}{=} f \circ h$ the obvious inhabitant, and given $f : X \to Y$ and $g, h : Y \to Z$ and $\alpha : g \overset{pw}{=} h$, we will denote by $\alpha(f) : g \circ f \overset{pw}{=} h \circ f$ the obvious inhabitant.

Lemma 0.2 (Naturality). *Let $\alpha : f \overset{pw}{=} g$. Then*

$$\prod_{x_1, x_2 : X} \prod_{\xi : x_1 = x_2} \Big(g(\xi) \bullet \alpha(x_1) = \alpha(x_2) \bullet f(\xi) \Big).$$

In a diagram:

$$\begin{tikzcd} f(x_1) \arrow[r, Rightarrow, "\alpha(x_1)"] \arrow[d, Rightarrow, "f(\xi)"] & g(x_1) \arrow[d, Rightarrow, "g(\xi)"] \\ f(x_2) \arrow[r, Rightarrow, "\alpha(x_2)"] & g(x_2) \end{tikzcd}$$

is commutative.

Proof. By path induction, it is enough to inhabit

$$\prod_{x:X} \Big(g(refl_x) \bullet \alpha(x) = \alpha(x) \bullet f(refl_x) \Big).$$

But $g(refl_x) :\equiv refl_{g(x)}$ and $f(refl_x) :\equiv refl_{f(x)}$, so this becomes the question of inhabiting $\alpha(x) = \alpha(x)$, which we can do with $refl_{\alpha(x)}$. □

Remark 0.3 (Iterated pointwise identities). Notice that $f(x) = g(x)$ is itself an X-parametrized type. Therefore, we can iterate and, given $\alpha, \beta : f \overset{pw}{=} g$, speak about $\alpha \overset{pw}{=} \beta$, etc.

1. The various variants of invertibility

1.1. **Various inverses.** Let $f : X \to Y$. Our goal is to study the property of f being an *equivalence*. This is the term corresponding to a homotopy equivalence (in the space interpretation) or an equivalence of categories (in the ∞-groupoid interpretation).

We can speak about left and right inverses:

$$LInv(f) \quad :\equiv \quad \sum_{g : Y \to X} \Big(id_X \overset{pw}{=} g \circ f \Big),$$

$$RInv(f) \quad :\equiv \quad \sum_{g:Y\to X} \left(f \circ g \overset{pw}{=} id_Y \right).$$

We can also speak about a two-sided inverse, which we call a "quasi-inverse" (because it is not really a good notion, more about this later) and about a pair consisting of a left inverse and a right inverse:

$$QInv(f) \quad :\equiv \quad \sum_{g:Y\to X} \left((id_X \overset{pw}{=} g \circ f) \times (f \circ g \overset{pw}{=} id_Y) \right),$$

$$LRInv(f) \quad :\equiv \quad LInv(f) \times RInv(f).$$

We also have an yet another type of inverse datum, that of a half-adjoint inverse. For $(g, \epsilon_l, \epsilon_r) : QInv(f)$ denote by

$$haattempt_{f,g,\epsilon_l,\epsilon_r} \quad :\equiv \quad \epsilon_r(f) \bullet f(\epsilon_l) : f \overset{pw}{=} f.$$

In a diagram, this is the concatenation:

$$f(x) \xRightarrow{f(\epsilon_l(x))} f(g(f(x))) \xRightarrow{\epsilon_r(f(x))} f(x)\ .$$

We then define

$$HAInv(f) \quad :\equiv \quad \sum_{(g,\epsilon_l,\epsilon_r):QInv(f)} \left(haattempt_{f,g,\epsilon_l,\epsilon_r} \overset{pw}{=} refl_f^{pw} \right).$$

1.2. **Equivalence.** Let $X, Y : Types$. For $f : X \to Y$ we define

$$IsEquiv(f) \quad :\equiv \quad LRInv(f).$$

We will see later that one could replace this also by $HAInv(f)$ or $IsContrFib(f)$, obtaining "equivalent" results. We define

$$X \cong Y \quad :\equiv \quad \sum_{f:X\to Y} IsEquiv(f).$$

Equivalence is an equivalence relation in an obvious sense.

By path induction, one easily inhabits

$$idtoequiv : \prod_{X,Y:Types} \left((X = Y) \to (X \cong Y) \right).$$

Voevodsky concieved the following axiom:

Axiom 1.1 (Univalence)**.** One inhabits

$$\prod_{X,Y:Types} IsEquiv(idtoequiv_{X,Y}).$$

2. Recollections 2: Truncated types

2.1. The hierarchy of truncatedness. For a type X, we define

$$IsTrunc_{-2}(X) \quad :\equiv \quad \sum_{c:X} \prod_{x:X} \Big(c = x \Big).$$

If we have defined already $IsTrunc_n(X)$ for some $n \in \mathbb{Z}_{\geqslant -2}$, we define

$$IsTrunc_{n+1}(X) \quad :\equiv \quad \prod_{x_1, x_2 : X} IsTrunc_n(x_1 = x_2).$$

A type X is said to be n-truncated if $IsTrunc_n(X)$ is inhabited. To be (-2)-truncated we also call to be contractible, and write $IsContr(X) :\equiv IsTrunc_{-2}(X)$, and to be (-1)-truncated we also call to be a mere proposition, and write $IsProp(X) :\equiv IsTrunc_{-1}(X)$. In the world of analogies with ∞-groupoids, to be 0-truncated should be thought of as being a set, to be 1-truncated as being a 1-groupoid, and so on.

I omit for now various things one can say about contractible types and mere propositions (obviously, look in the book). Maybe one thing to record is that given a type X, one axiomatically introduces the type $||X||$, which is a mere proposition (the "propositional reduction" of X). Thus, intuitively, if X is non-empty then $||X||$ is contractible, and if X is empty then $||X||$ is empty.

Also, let us note that inhabiting a mere proposition might be called establishing it.

2.2. Contractible fibers. We will need the following terminologies. Let X, Y be types and let $f : X \to Y$. For $y : Y$ we define

$$Fib_f(y) \quad :\equiv \quad \sum_{x:X} \Big(f(x) = y \Big)$$

and

$$IsContrFib(f) \quad :\equiv \quad \prod_{y:Y} IsContr(Fib_f(y)).$$

3. Plan

We will inhabit the following function types in the following sections:

$$\begin{tikzcd}
& IsContrFib(f) \arrow[d] & . \\
LRInv(f) \arrow[r, bend left] & QInv(f) \arrow[l, bend left] \arrow[r, bend left] & HAInv(f) \arrow[l, bend left] \arrow[ul, bend right]
\end{tikzcd}$$

Notice that inhabiting $QInv(f) \to LRInv(f)$ and $HAInv(f) \to QInv(f)$ is clear.

We will also show that the types

$$IsContrFib(f), LRInv(f), HAInv(f)$$

are all mere propositions. Thus, since the above diagram is connected in the directed sense, we deduce that the types

$$IsContrFib(f), LRInv(f), HAInv(f)$$

are all equivalent mere propositions. Thus, in principle, each one of them could serve as $IsEquiv(f)$.

We will also explain that $QInv(f)$ is not a mere proposition, in general. In other words, $||QInv(f)||$ is equivalent to $IsEquiv(f)$, but $QInv(f)$ is not, in general.

4. Inhabiting $LRInv(f) \to QInv(f)$

Lemma 4.1. *Let $f : X \to Y$ and let $LRInv(f)$. Then $QInv(f)$.*

Proof. We denote the given inhabiting element $(g_l, g_r, \epsilon_l, \epsilon_r) : LRInv(f)$ as usual. Then we inhabit[3]

$$f \circ g_l \overset{pw}{=} f \circ g_l \circ id_Y \overset{pw}{=} f \circ g_l \circ f \circ g_r \overset{pw}{=} f \circ id_X \circ g_r \overset{pw}{=} f \circ g_r \overset{pw}{=} id_Y .$$

So we have $g_l : Y \to X$ and inhabitants of $id_X \overset{pw}{=} g_l \circ f$ and of $f \circ g_l \overset{pw}{=} id_Y$, therefore an inhabitant of $QInv(f)$. $\square$

5. Inhabiting $QInv(f) \to HAInv(f)$

Proposition 5.1. *Let $f : X \to Y$ and let $QInv(f)$. Then $HAInv(f)$.*

Proof. We denote the given inhabiting element $(g, \epsilon_l, \epsilon_r) : QInv(f)$ as usual.

Given $\alpha : id_X \overset{pw}{=} id_X$, let us redefine

$$\epsilon_l' \quad :\equiv \quad \epsilon_l \bullet \alpha.$$

Then the new $haattempt_{f,g,\epsilon_l',\epsilon_r}$ will be the concatenation

$$\begin{tikzcd} f(x) \arrow[r, Rightarrow, "f(\epsilon_l(x))"] & f(g(f(x))) \arrow[r, Rightarrow, "\epsilon_r(f(x))"] & f(x) . \\ f(x) \arrow[u, Rightarrow, "f(\alpha(x))"] & & \end{tikzcd}$$

Therefore, if we can construct α for which $f(\alpha) \overset{pw}{=} haattepmt^{-1}_{f,g,\epsilon_l,\epsilon_r}$, we see that $haattempt_{f,g,\epsilon_l',\epsilon_r}$ will be pointwise identified with $refl^{pw}_f$, as desired. The construction of such α is given by Lemma 5.2 that follows. $\square$

Lemma 5.2. *Let $f : X \to Y$ and let $QInv(f)$ and let $\beta : f \overset{pw}{=} f$. Then we construct $\alpha : id_X \overset{pw}{=} id_X$ and $f(\alpha) \overset{pw}{=} \beta$.*

Proof. We denote the given inhabiting element $(g, \epsilon_l, \epsilon_r) : QInv(f)$ as usual.

Let us first fix some $\gamma : f \overset{pw}{=} f$. We try to consider the following diagram with some $\alpha : id_X \overset{pw}{=} id_X$:

$$\begin{tikzcd} f(x) \arrow[r, Rightarrow, "f(\epsilon_l(x))"] \arrow[d, Rightarrow, "f(\alpha(x))"] & f(g(f(x))) \arrow[r, Rightarrow, "\epsilon_r(f(x))"] \arrow[d, Rightarrow, "f(g(\gamma(x)))"] & f(x) \arrow[d, Rightarrow, "\gamma(x)"] & . \\ f(x) \arrow[r, Rightarrow, "f(\epsilon_l(x))"] & f(g(f(x))) \arrow[r, Rightarrow, "\epsilon_r(f(x))"] & f(x) & \end{tikzcd}$$

---

[3]When we say that we inhabit something like $X = Y = Z$, we mean that we inhabit $X = Y$, and inhabit $Y = Z$, and then inhabit $X = Z$ using concatenation, etc.

The right square commutes by Lemma 0.2, while the left square will commute if we set

$$\alpha \quad :\equiv \quad \epsilon_l^{-1} \bullet g(\gamma) \bullet \epsilon_l : id_X \overset{pw}{=} id_X.$$

Then, for this choice of α, we obtain

$$f(\alpha) \overset{pw}{=} haattempt_{f,g,\epsilon_l,\epsilon_r}^{-1} \bullet \gamma \bullet haattempt_{f,g,\epsilon_l,\epsilon_r}.$$

Thus, setting

$$\gamma \quad :\equiv \quad haattempt_{f,g,\epsilon_l,\epsilon_r} \bullet \beta \bullet haattempt_{f,g,\epsilon_l,\epsilon_r}^{-1} : f \overset{pw}{=} f,$$

we will obtain $f(\alpha) \overset{pw}{=} \beta$.

Alternatively, we can use Lemma 9.1.

$\square$

6. Inhabiting $IsContrFib(f) \to QInv(f)$

Proposition 6.1. *Let $f : X \to Y$ and let $IsContrFib(f)$. Then $QInv(f)$.*

Proof. We have

$$IsContrFib(f) \cong IsContr\left(\prod_{y:Y} Fib_f(y)\right).$$

Therefore, assuming the former, we obtain a term $\widetilde{g} : \prod_{y:Y} Fib_f(y)$. Write $\widetilde{g}(y) \equiv: (g(y), \beta(y))$ so $\beta(y) : f(g(y)) = y$. Then $f \circ g \overset{pw}{=} id_Y$. For the other direction, we consider $\widetilde{g} \circ f : \prod_{x:X} Fib_f(f(x))$. We also have another element $taut : \prod_{x:X} Fib_f(f(x))$ sending $taut(x) :\equiv (x, refl_{f(x)})$. Since this type is contractible, we obtain an inhabitant of $\widetilde{g} \circ f = taut$, so in particular of $\widetilde{g} \circ f \overset{pw}{=} taut$. Composing with

$$\prod_{x:X} Fib_f(f(x)) \to \prod_{x:X} X$$

inhabits $g \circ f \overset{pw}{=} id_X$. $\square$

7. Inhabiting $HAInv(f) \to IsContrFib(f)$

First, we will see how to manage identity types in fibers.

Lemma 7.1. *Let $f : X \to Y$, and let $y : Y$ and let $(x_1, \beta_1), (x_2, \beta_2) : Fib_f(y)$. Then*

$$\Big((x_1, \beta_1) = (x_2, \beta_2)\Big) \cong \left(\sum_{\alpha : x_1 = x_2} \beta_2 \bullet f(\alpha) = \beta_1\right).$$

In a diagram, given

$$\begin{tikzcd}
x_1 & & x_2 \\
f(x_1) \arrow[dr, Rightarrow, "\beta_1"'] & & f(x_2) \arrow[dl, Rightarrow, "\beta_2"] \\
& y &
\end{tikzcd},$$

to inhabit the identity type $(x_1, \beta_1) = (x_2, \beta_2)$ we need to complement by an α and a commutation of the lower triangle:

$$\begin{tikzcd} x_1 \arrow[rr, Rightarrow, "\alpha"] & & x_2 \end{tikzcd} \quad .$$

$$\begin{tikzcd} f(x_1) \arrow[rr, Rightarrow, "f(\alpha)"] \arrow[dr, Rightarrow, "\beta_1"'] & & f(x_2) \arrow[dl, Rightarrow, "\beta_2"] \\ & y & \end{tikzcd}$$

Proof (sketch). Earlier in the book, there was a description of identity types in sum types, which gives here

$$\Big((x_1, \beta_1) = (x_2, \beta_2)\Big) \cong \sum_{\alpha : x_1 = x_2} \Big(trans_\alpha(\beta_1) = \beta_2\Big).$$

One also inhabits, via path induction and so on,

$$\Big(trans_\alpha(\beta_1) = \beta_2\Big) \cong \Big(\beta_2 \bullet f(\alpha) = \beta_1\Big),$$

and summing this up we obtain the desired inhabitation. $\square$

Proposition 7.2. *Let $f : X \to Y$ and let $HAInv(f)$. Then $IsContrFib(f)$.*

Proof. Denote by $(g, \epsilon_l, \epsilon_r, \delta) : HAInv(f)$ the given inhabitant (so $g : Y \to X$, $\epsilon_l : id_X \overset{pw}{=} g \circ f, \epsilon_r : f \circ g \overset{pw}{=} id_Y$ and $\delta : haattempt_{f,g,\epsilon_l,\epsilon_r} \overset{pw}{=} refl_f^{pw}$) and fix $y : Y$.

Notice that $Fib_f(y)$ is inhabited by $(g(y), \epsilon_r(y))$. Since one has $C \times IsProp(C) \to IsContr(C)$, it is left to show $IsProp(Fib_f(Y))$. So fix $(x_1, \beta_1), (x_2, \beta_2) : Fib_f(y)$ and we need to inhabit $(x_1, \beta_1) = (x_2, \beta_2)$. By using Lemma 7.1, it is enough to find an inhabitant $\alpha : x_1 = x_2$ and then to inhabit $\beta_2 \bullet f(\alpha) = \beta_1$.

We have

$$\begin{tikzcd} x_1 & & x_2 \end{tikzcd} \quad .$$

$$\begin{tikzcd} f(x_1) \arrow[dr, Rightarrow, "\beta_1"'] & & f(x_2) \arrow[dl, Rightarrow, "\beta_2"] \\ & y & \end{tikzcd}$$

We then have

$$\begin{tikzcd} x_1 \arrow[d, Rightarrow, "\epsilon_l(x_1)"'] & & x_2 \arrow[d, Rightarrow, "\epsilon_l(x_2)"] \\ g(f(x_1)) \arrow[dr, Rightarrow, "g(\beta_1)"'] & & g(f(x_2)) \arrow[dl, Rightarrow, "g(\beta_2)"] \\ & g(y) & \end{tikzcd}$$

and therefore composing we obtain an inhabitant $\alpha : x_1 = x_2$. Next, we apply f to this diagram and append an identity, obtaining

$$\begin{tikzcd}
f(x_1) \arrow[rr, Rightarrow, "f(\alpha)"] \arrow[d, Rightarrow, "f(\epsilon_l(x_1))"'] & & f(x_2) \arrow[d, Rightarrow, "f(\epsilon_l(x_2))"] \\
f(g(f(x_1))) \arrow[dr, Rightarrow, "f(g(\beta_1))"'] & & f(g(f(x_2))) \arrow[dl, Rightarrow, "f(g(\beta_2))"] \\
& f(g(y)) \arrow[d, Rightarrow, "\epsilon_r(y)"] & \\
& y &
\end{tikzcd} .$$

Since by definition of α the diagram commutes (a 2-homotopy! i.e. an inhabitant of the identity type between two inhabitants of the identity type $f(x_1) = y$), it is enough to find an inhabitant of the identity type between the two inhabitants of $f(x_2) = y$, one given by the right route in the diagram, and the other given by β_2. An analogous procedure will then identify the left route with β_1, and we will therefore find an inhabitant of $\beta_2 \bullet f(\alpha) = \beta_1$, as desired.

To that end, consider now the right route in the diagram and add to it an appendage:

$$\begin{tikzcd}
& f(x_2) \arrow[d, Rightarrow, "f(\epsilon_l(x_2))"] \\
& f(g(f(x_2))) \arrow[dl, Rightarrow, "f(g(\beta_2))"'] \arrow[d, Rightarrow, "\epsilon_r(f(x_2))"] \\
f(g(y)) \arrow[d, Rightarrow, "\epsilon_r(y)"] & f(x_2) \arrow[dl, Rightarrow, "\beta_2"] \\
y &
\end{tikzcd} .$$

Notice that the parallelogram commutes by Lemma 0.2. Furthermore, we have an identification of the composition of the upper right two arrows with the reflexive inhabitant, by definition of an half-adjoint inverse. From this we obtain the desired. $\square$

8. Establishing $IsProp(LRInv(f))$

Lemma 8.1. *Let $f : X \to Y$. Define*

$$c_f : (Y \to X) \to (X \to X)$$

by

$$c_f(g) :\equiv g \circ f.$$

Then

$$QInv(f) \to QInv(c_f).$$

Also,

$$Fib_{c_f}(id_X) \cong LInv(f).$$

Proof. The first claim is easy.

As for the second claim, we have

$$Fib_{c_f}(id_X) :\equiv \sum_{g:Y \to X} \left(g \circ f = id_X \right) \cong$$

and by function extensionality we continue

$$\cong \sum_{g:Y \to X} \left(g \circ f \overset{pw}{=} id_X \right) \equiv: LInv(f).$$

□

Lemma 8.2. *Let $f : X \to Y$ and let $QInv(f)$. Then $IsContr(LInv(f))$ and $IsContr(RInv(f))$, or equivalently $IsContr(LRInv(f))$.*

Proof. Let us show $QInv(f) \to IsContr(LInv(f))$ (the other check is analogous). Let us consider

$$c_f : (Y \to X) \to (X \to X)$$

as in the preceding Lemma. Then by this Lemma we have

$$QInv(f) \to QInv(c_f).$$

By Proposition 5.1 we have

$$QInv(c_f) \to HAInv(c_f).$$

By Proposition 7.2 we have

$$HAInv(c_f) \to IsContr(Fib_{c_f}(id_X)).$$

By the preceding Lemma we have

$$IsContr(Fib_{c_f}(id_X)) \to IsContr(LInv(f)).$$

Composing all those together, we obtain the desired. □

Proposition 8.3. *Let $f : X \to Y$. Then*

$$IsProp(LRInv(f)).$$

Proof. We have, for a type Z,

$$IsProp(Z) \cong \big(Z \to IsContr(Z)\big),$$

So it suffices to assume that $LRInv(f)$ is inhabited and show that $IsContr(LRInv(f))$. But if $LRInv(f)$ is inhabited, then so is $QInv(f)$, and therefore so is $IsContr(LRInv(f))$ by Lemma 8.2. □

9. Establishing $IsProp(HAInv(f))$

Lemma 9.1.

(1) *Let Y, Z : Tpyes and let $f : Y \to Z$ and suppose that f is an equivalence. Then for all $y_1, y_2, : Y$ the function in*

$$(y_1 = y_2) \to (f(y_1) = f(y_2))$$

given by

$$\alpha \mapsto f(\alpha)$$

is an equivalence.

(2) *Let $Y, Z : Types$ and let $f : Y \to Z$ and suppose that f is an equivalence. Let $X : Types$. Then for all $f_1, f_2 : X \to Y$ the function in*

$$(f_1 \overset{pw}{=} f_2) \to (f \circ f_1 \overset{pw}{=} f \circ f_2)$$

given by

$$\Big(\alpha_x : f_1(x) = f_2(x)\Big)_{x:X} \mapsto \Big(f(\alpha_x) : f(f_1(x)) = f(f_2(x))\Big)_{x:X}$$

is an equivalence.

Lemma 9.2.

(1) *Let $Y : Types$ and let $y_1, y_2 : Y$ and let $\beta : y_1 = y_2$. Then for all $y : Y$ the function*

$$(y = y_1) \to (y = y_2)$$

given by

$$\alpha \mapsto \beta \bullet \alpha$$

is an equivalence.

(2) *Let $X, Y : Types$ and let $f_1, f_2 : X \to Y$ and let $\beta : f_1 \overset{pw}{=} f_2$. Then for all $f : X \to Y$ the function*

$$(f \overset{pw}{=} f_1) \to (f \overset{pw}{=} f_2)$$

given by

$$\Big(\alpha_x : f(x) = f_1(x)\Big)_{x:X} \mapsto \Big(\beta_x \bullet \alpha_x : f(x) = f_2(x)\Big)_{x:X}$$

is an equivalence.

Lemma 9.3. *Let $f : X \to Y$ and let $(g, \epsilon_l, \epsilon_r, \delta) : HAInv(f)$. Then the function*

$$adj : (id_X \overset{pw}{=} g \circ f) \to (f \overset{pw}{=} f)$$

given by sending ϵ to $haattempt_{f,g,\epsilon,\epsilon_r}$ is an equivalence.

Proof. Thus, adj is the composition of the function in

$$(id_X \overset{pw}{=} g \circ f) \to (f \overset{pw}{=} f \circ g \circ f)$$

sending

$$\Big(\alpha_x : x = g(f(x))\Big)_{x:X} \mapsto \Big(f(\alpha_x) : f(x) = f(g(f(x)))\Big)_{x:X}$$

and the function in

$$(f \overset{pw}{=} f \circ g \circ f) \to (f \overset{pw}{=} f)$$

sending

$$\Big(\beta_x : f(x) = f(g(f(x)))\Big)_{x:X} \mapsto \Big((\epsilon_r)_{f(x)} \bullet \beta_x\Big)_{x:X}.$$

So, it is enough to check that each of those functions is an equivalence. The first one is an equivalence by Lemma 9.1 and the second one is an equivalence by Lemma 9.2. $\square$

Lemma 9.4. *Let $f : X \to Y$ and let $(g, \epsilon_l, \epsilon_r) : QInv(f)$. Denote by*

$$frgt : HAInv(f) \to RInv(f)$$

the function which forgets information in the obvious way, and denote by

$$adj : (id_X \overset{pw}{=} g \circ f) \to (f \overset{pw}{=} f)$$

the function which sends ϵ to $haattempt_{f,g,\epsilon,\epsilon_r}$. Then one has

$$Fib_{frgt}(g, \epsilon_r) \cong Fib_{adj}(refl_f^{pw}).$$

Proof. This is clear from Lemma 7.1. $\square$

Proposition 9.5. *Let $X, Y : Types$ and let $f : X \to Y$. Then $HAInv(f)$ is a mere proposition.*

Proof. It is enough to assume that $HAInv(f)$ is inhabited, and show that $HAInv(f)$ is contractible. We consider the forgetting function $frgt : HAInv(f) \to RInv(f)$. Since $HAInv(f)$ is inhabited, $QInv(f)$ is inhabited as well, and thus by Lemma 8.2 we have that $RInv(f)$ is contractible. Therefore, in order to show that $HAInv(f)$ is contractible, it is enough to show that $frgt$ is an equivalence. We saw that this is logically the same as showing that the fibers of $frgt$ are contractible, for which by Lemma 9.4 it is enough to show that all fibers of the function adj are contractible, which again is equivalent to adj being an equivalence, which follows from Lemma 9.3. $\square$

10. $QInv(f)$ is not a mere proposition in general

Lemma 10.1. *Let $f : X \to Y$ and let $QInv(f)$. Then the fibers of the forgetful function $frgt : QInv(f) \to RInv(f)$ are all equivalent to the type $id_X \overset{pw}{=} id_X$.*

Proof. The fiber of $frgt$ over some $(g, \epsilon_r) : RInv(f)$ is equivalent to $id_X \overset{pw}{=} g \circ f$. This type is inhabited since $QInv(f)$ is inhabited. Using Lemma 9.2 we have an equivalence between $id_X \overset{pw}{=} id_X$ and $id_X \overset{pw}{=} g \circ f$. $\square$

Remark 10.2. Recall that we saw that if $QInv(f)$ is inhabited then $RInv(f)$ is contractible. Therefore, to exhibit f for which $QInv(f)$ is not contractible, it is enough to exhibit f for which $QInv(f)$ is inhabited and $id_X \overset{pw}{=} id_X$ is not contractible (then, in view of Lemma 10.1, $QInv(f)$ can't be contractible because otherwise the fibers of $frgt : QInv(f) \to RInv(f)$ would be contractible). Taking simply $f :\equiv id_X$, it is enough to exhibit a type X for which $id_X \overset{pw}{=} id_X$ is not contractible. Intuitively, the classifying space of $Aut(\mathbb{2})$ should be such a type, and this can be checked, as below.

Lemma 10.3. *Let $X : Types$ be 1-truncated and connected. Let $x : X$. Then*

$$(id_X \overset{pw}{=} id_X) \cong \sum_{\alpha : x = x} \prod_{\beta : x = x} \Big(\alpha \bullet \beta = \beta \bullet \alpha\Big).$$

Lemma 10.4. *The type $\mathbb{2} = \mathbb{2}$ is not connected, and the forgetting function in*

$$\left(\sum_{\alpha : \mathbb{2} = \mathbb{2}} \prod_{\beta : \mathbb{2} = \mathbb{2}} \left(\alpha \bullet \beta = \beta \bullet \alpha \right) \right) \to \left(\mathbb{2} = \mathbb{2} \right)$$

is an equivalence.

Corollary 10.5. *Consider the type*

$$X \quad :\equiv \quad \sum_{Y : Types} ||\mathbb{2} = Y||.$$

Then X is 1-truncated and connected, and $(id_X \overset{pw}{=} id_X) \cong \left(\mathbb{2} = \mathbb{2} \right)$, so is not connected, and in particular not contractible.

11. Straightening-Unstraightening (Grothendieck construction)

Proposition 11.1. *Let $X : Types$. One has*

$$\left(\sum_{Y : Types} (Y \to X) \right) \cong \left(X \to Types \right).$$

Proof. Given

$$\Phi \equiv (Y, \pi) : \sum_{Y : Types} (Y \to X)$$

we construct

$$\Psi : X \to Types$$

by sending $x : X$ to $Fib_\pi(x)$. We claim that this is an equivalence. We construct an inverse - given Ψ, we construct Φ by considering $Y :\equiv \sum_{x:X} \Psi(x)$ and $\pi :\equiv pr_1$. We have now maps

$$\begin{tikzcd} \left(\sum_{Y:Types} (Y \to X) \right) \arrow[r, bend left, "str"] & \left(X \to Types \right) \arrow[l, bend left, "unstr"] \end{tikzcd}$$

and we need to construct homotopies of compositions to identities.

The element $(unstr \circ str)(\Phi)$ is given by

$$\left(\sum_{x:X} Fib_\pi(x), pr_1 \right).$$

We have

$$\sum_{x:X} Fib_\pi(x) \equiv \left(\sum_{x:X} \sum_{y:Y} (\pi(y) = x) \right) = \sum_{y:Y} \sum_{x:X} (\pi(y) = x) = \sum_{y:Y} (\text{something contractible}) = Y.$$

One checks that the transport along that of pr_1 can be identified with π, giving the desired homotopy.

Conversely, the element $(str \circ unstr)(\Psi)$ is given by sending $x : X$ to the fiber of $pr_1 : \left(\sum_{x:X} \Psi(x) \right) \to X$ over $x : X$. This can be identified with $\Psi(x)$.

∎