HoTT Reading Notes

Pierre-Yves Gaillard

Tuesday 10$^{\text{th}}$ October, 2017, 21:12

This text is available at

http://iecl.univ-lorraine.fr/∼Pierre-Yves.Gaillard/HoTT/ReadingNotes

https://github.com/Pierre-Yves-Gaillard/HoTT_Reading_Notes

http://goo.gl/Dz6r8g

This is an informal set of comments on the HoTT Book:

http://homotopytypetheory.org/book

This is a work in progress. John Dougherty's text

https://github.com/jdoughertyii/hott-exercises

has been most helpful.

Contents

1 Chapter 1

1.1 Path induction implies based path induction, §1.12.2

Let us show that path induction implies based path induction. Let A be a type, let $a : A$, let

$$C_0 : \prod_{x:A} (a = x) \to \mathcal{U},$$

and let $c_0 : C_0(a, \mathsf{refl}_a)$. We must find an

$$f : \prod_{x:A} \prod_{p:a=x} C_0(x, p) \tag{1}$$

such that

$$f(a, \mathsf{refl}_a) \equiv c_0. \tag{2}$$

Define $D : \prod_{x,y:A}(x = y) \to \mathcal{U}$ by

$$D(x, y, p) :\equiv \prod_{C:\prod_{z:A}(x=z)\to\mathcal{U}} C(x, \mathsf{refl}_x) \to C(y, p) \tag{3}$$

and $d : \prod_{x:A} D(x, x, \mathsf{refl}_x)$ by

$$d(x, C, c) :\equiv c. \tag{4}$$

By path induction we get a

$$g : \prod_{x,y:A} \prod_{p:x=y} D(x,y,p) \tag{5}$$

such that

$$g(x,x,\mathsf{refl}_x) \equiv d(x) \tag{6}$$

for all $x : A$. We set

$$f(x,p) :\equiv g(a,x,p,C_0,c_0) \tag{7}$$

for $x : A$ and $p : a = x$.

We prove (1). Let $x : A$, $p : a = x$. We must show $f(x,p) : C_0(x,p)$. By (7) it suffices to show

$$g(a,x,p,C_0,c_0) : C_0(x,p). \tag{8}$$

Judgment (5) implies $g(a,x,p) : D(a,x,p)$. Now (3) implies (8). This proves (1).

To prove (2), observe

$$f(a,\mathsf{refl}_a) \overset{\text{(a)}}{\equiv} g(a,a,\mathsf{refl}_a,C_0,c_0) \overset{\text{(b)}}{\equiv} d(a,C_0,c_0) \overset{\text{(c)}}{\equiv} c_0,$$

where (a) follows from (7), (b) follows from (6), and (c) follows from (4). □

1.2 Exercise 1.3

Statement. Derive the induction principle for products $\mathsf{ind}_{A\times B}$, using only the projections and the propositional uniqueness principle $\mathsf{uniq}_{A\times B}$. Verify that the definitional equalities are valid. Generalize $\mathsf{uniq}_{A\times B}$ to Σ-types, and do the same for Σ-types. *(This requires concepts from Chapter 2.)*

Solution. It suffices to handle Σ-types.

(a) We assume that, for all type family $B : A \to \mathcal{U}$ we have a map

$$\mathsf{pr}_1 : \left(\sum_{a:A} B(a)\right) \to A$$

and a dependent function

$$\mathsf{pr}_2 : \prod_{x:\sum_{a:A} B(a)} B(\mathsf{pr}_1(x))$$

satisfying

$$\mathsf{pr}_1(a,b) \equiv a, \quad \mathsf{pr}_2(a,b) \equiv b \tag{9}$$

for all $a : A$ and all $b : B(a)$.

(b) We also assume that, for all type family $B : A \to \mathcal{U}$ we have a dependent function

$$\mathsf{uniq}_{\sum_{a:A} B(a)} : \prod_{x:\sum_{a:A} B(a)} \big((\mathsf{pr}_1(x), \mathsf{pr}_2(x)) = x\big)$$

such that

$$\mathsf{uniq}_{\sum_{a:A} B(a)}(a,b) \equiv \mathsf{refl}_{(a,b)} \tag{10}$$

for all $a : A$ and all $b : B(a)$.

(c) We define

$$\mathsf{ind}_{\sum_{a:A} B(a)} : \prod_{C:\left(\sum_{a:A} B(a)\right)\to\mathcal{U}} \left(\prod_{a:A} \prod_{b:B(a)} C((a,b))\right) \to \prod_{x:\sum_{a:A} B(a)} C(x)$$

thanks to the transport principle, which is indeed a concept from Chapter 2, by

$$\mathsf{ind}_{\sum_{a:A} B(a)}(C,g,x) :\equiv \Big(\mathsf{uniq}_{\sum_{a:A} B(a)}(x)\Big)_\star \Big(g\big(\mathsf{pr}_1(x)\big)\big(\mathsf{pr}_2(x)\big)\Big). \tag{11}$$

(d) We must prove

$$\mathsf{ind}_{\sum_{a:A} B(a)}(C,g,(a,b)) \equiv g(a)(b)$$

for all $a : A$ and all $b : B(a)$. But this follows from (9), (10) and (11). □

1.3 Exercise 1.5

Statement. Show that if we define $A + B :\equiv \sum_{x:\mathbf{2}} \mathsf{rec}_{\mathbf{2}}(\mathcal{U}, A, B, x)$, then we can give a definition of ind_{A+B} for which the definitional equalities stated in §1.7 hold.

Solution. Recall that we have

$$\mathsf{rec}_{\mathbf{2}} : \prod_{C:\mathcal{U}} C \to C \to \mathbf{2} \to C, \quad \mathsf{rec}_{\mathbf{2}}(C, c_0, c_1, 0_{\mathbf{2}}) :\equiv c_0, \quad \mathsf{rec}_{\mathbf{2}}(C, c_0, c_1, 1_{\mathbf{2}}) :\equiv c_1.$$

We define $\mathsf{inl} : A \to A + B$ by $\mathsf{inl}(a) :\equiv (0_{\mathbf{2}}, a)$, and we define inr similarly. Let $C : (A + B) \to \mathcal{U}$ be given, and put

$$A' :\equiv \prod_{a:A} C(\mathsf{inl}(a)), \quad B' :\equiv \prod_{b:B} C(\mathsf{inr}(b)).$$

We must define

$$\mathsf{ind}_{A+B}(C) : A' \to B' \to \prod_{x:A+B} C(x).$$

Let $g_0 : A'$, $g_1 : B'$ be given. We must define

$$\mathsf{ind}_{A+B}(C, g_0, g_1) : \prod_{x:A+B} C(x).$$

Set $T :\equiv \mathsf{rec}_{\mathbf{2}}(\mathcal{U}, A, B) : \mathbf{2} \to \mathcal{U}$. We define $D : \mathbf{2} \to \mathcal{U}$ by

$$D(n) :\equiv \prod_{u:T(n)} C((n, u)).$$

Note that $D(0_{\mathbf{2}}) \equiv A'$ and $D(1_{\mathbf{2}}) \equiv B'$. Recall that

$$\mathsf{ind}_{\mathbf{2}}(D) : D(0_{\mathbf{2}}) \to D(1_{\mathbf{2}}) \to \prod_{n:\mathbf{2}} D(n).$$

In particular we have

$$\mathsf{ind}_{\mathbf{2}}(D, g_0, g_1) : \prod_{n:\mathbf{2}} \prod_{u:T(n)} C((n, u))$$

with

$$\mathsf{ind}_{\mathbf{2}}(D, g_0, g_1)(0_{\mathbf{2}}) :\equiv g_0, \quad \mathsf{ind}_{\mathbf{2}}(D, g_0, g_1)(1_{\mathbf{2}}) :\equiv g_1.$$

Since

$$\mathsf{ind}_{\sum_{n:\mathbf{2}} T(n)}(C) : \left(\prod_{n:\mathbf{2}} \prod_{u:T(n)} C((n, u)) \right) \to \prod_{x:A+B} C(x),$$

we can put

$$\mathsf{ind}_{A+B}(C, g_0, g_1) :\equiv \mathsf{ind}_{\sum_{n:\mathbf{2}} T(n)}(C, \mathsf{ind}_{\mathbf{2}}(D, g_0, g_1)).$$

We must check

$$\mathsf{ind}_{A+B}(C, g_0, g_1, (0_{\mathbf{2}}, a)) \equiv g_0(a), \tag{12}$$

$$\mathsf{ind}_{A+B}(C, g_0, g_1, (1_{\mathbf{2}}, b)) \equiv g_1(b). \tag{13}$$

We have

$$\begin{aligned} \mathsf{ind}_{A+B}(C, g_0, g_1, (0_{\mathbf{2}}, a)) &\equiv \mathsf{ind}_{\sum_{n:\mathbf{2}} T(n)}\big(C, \mathsf{ind}_{\mathbf{2}}(D, g_0, g_1), (0_{\mathbf{2}}, a)\big) \\ &\equiv \mathsf{ind}_{\mathbf{2}}(D, g_0, g_1)(0_{\mathbf{2}})(a) \\ &\equiv g_0(a), \end{aligned}$$

the three definitional equalities following respectively from the definitions of the dependent functions

$$\mathsf{ind}_{A+B}, \quad \mathsf{ind}_{\sum_{n:\mathbf{2}} T(n)}, \quad \mathsf{ind}_{\mathbf{2}}.$$

This proves (12). The proof of (13) is similar. □

1.4 Exercise 1.6

Statement. Show that if we define $A \times B :\equiv \prod_{x:\mathbf{2}} \mathsf{rec}_{\mathbf{2}}(\mathcal{U}, A, B, x)$, then we can give a definition of $\mathsf{ind}_{A\times B}$ for which the definitional equalities stated in §1.5 hold propositionally (i.e. using equality types). *(This requires the function extensionality axiom, which is introduced in §2.9.)*

Solution. As above set $T :\equiv \mathsf{rec}_{\mathbf{2}}(\mathcal{U}, A, B)$, so that $T : \mathbf{2} \to \mathcal{U}$ and

$$A \times B :\equiv \prod_{n:\mathbf{2}} T(n).$$

Recall that we have

$$\mathsf{ind}_{\mathbf{2}}(T) : A \to B \to A \times B.$$

For $a{:}A,\ b{:}B$ we abbreviate $\mathsf{ind}_{\mathbf{2}}(T, a, b)$ by (a, b); in particular we have

$$(a, b) : A \times B, \quad (a, b)(0_{\mathbf{2}}) \equiv a, \quad (a, b)(1_{\mathbf{2}}) \equiv b.$$

Let $C : (A \times B) \to \mathcal{U}$. We must define

$$\mathsf{ind}_{A\times B}(C) : \left(\prod_{a:A} \prod_{b:B} C((a, b))\right) \to \prod_{u:A\times B} C(u).$$

Let $u : A \times B$ and define $D : \mathbf{2} \to \mathcal{U}$ by

$$D(n) :\equiv \Big(\big(u(0_{\mathbf{2}}), u(1_{\mathbf{2}})\big)(n) = u(n)\Big),$$

and recall that

$$\mathsf{ind}_{\mathbf{2}}(D) : D(0_{\mathbf{2}}) \to D(1_{\mathbf{2}}) \to \prod_{n:\mathbf{2}} D(n),$$

and that

$$\mathsf{funext} : \left(\prod_{n:\mathbf{2}} D(n)\right) \to \Big(\big(u(0_{\mathbf{2}}), u(1_{\mathbf{2}})\big) = u\Big).$$

Setting $\alpha :\equiv \mathsf{ind}_{\mathbf{2}}(D, \mathsf{refl}_{u(0_{\mathbf{2}})}, \mathsf{refl}_{u(1_{\mathbf{2}})})$, we get

$$\alpha : \prod_{n:\mathbf{2}} D(n), \quad \mathsf{funext}(\alpha) : \big(u(0_{\mathbf{2}}), u(1_{\mathbf{2}})\big) = u,$$

$$\mathsf{funext}(\alpha)_\star : C\Big(\big(u(0_{\mathbf{2}}), u(1_{\mathbf{2}})\big)\Big) \to C(u).$$

In particular we can set

$$\mathsf{ind}_{A\times B}(C, g, u) :\equiv \mathsf{funext}(\alpha)_\star\Big(g\big(u(0_{\mathbf{2}})\big)\big(u(1_{\mathbf{2}})\big)\Big).$$

To show that the definitional equalities stated in §1.5 hold propositionally, we assume $u \equiv (a, b)$. In particular we have

$$\alpha : \prod_{n:\mathbf{2}} \Big((a, b)(n) = (a, b)(n)\Big).$$

It is easy to prove $\alpha(n) = \mathsf{refl}_{(a,b)(n)}$ by induction. This gives

$$\mathsf{ind}_{A\times B}(C, g, (a, b)) = g(a)(b),$$

as required. □

1.5 Exercise 1.8

Statement. Define multiplication and exponentiation using $\mathsf{rec}_{\mathbb{N}}$. Verify that

$$(\mathbb{N}, +, 0, \times, 1)$$

is a semiring using only $\mathsf{ind}_{\mathbb{N}}$. You will probably also need to use symmetry and transitivity of equality, Lemmas 2.1.1 and 2.1.2.

Solution. We put

$$mn :\equiv \mathsf{rec}_{\mathbb{N}}(\ \mathbb{N} \to \mathbb{N}\ ,\ \lambda n.0\ ,\ \lambda m.\lambda g.\lambda n.n + g(n)\ ,\ m\ ,\ n\ ),$$

$$n^m :\equiv \mathsf{rec}_{\mathbb{N}}(\ \mathbb{N} \to \mathbb{N}\ ,\ \lambda n.1\ ,\ \lambda m.\lambda g.\lambda n.ng(n)\ ,\ m\ ,\ n\ ).$$

We omit the verification of the fact that $(\mathbb{N}, +, 0, \times, 1)$ is a semiring. □

For the reader's convenience we briefly recall the definition of the recursor for $\mathbb{N}$:

$$\mathsf{rec}_{\mathbb{N}} : \prod_{C:\mathcal{U}} C \to (\mathbb{N} \to C \to C) \to \mathbb{N} \to C,$$

$$\mathsf{rec}_{\mathbb{N}}(C, c_0, c_s, 0) :\equiv c_0,$$

$$\mathsf{rec}_{\mathbb{N}}(C, c_0, c_s, \mathsf{succ}(n)) :\equiv c_s(n, \mathsf{rec}_{\mathbb{N}}(C, c_0, c_s, n)).$$

1.6 Exercise 1.10

Statement. Show that the Ackermann function $\mathsf{ack} : \mathbb{N} \to \mathbb{N} \to \mathbb{N}$ is definable using only $\mathsf{rec}_{\mathbb{N}}$ satisfying the following equations:

$$\mathsf{ack}(0, n) \equiv \mathsf{succ}(n), \tag{14}$$

$$\mathsf{ack}(\mathsf{succ}(m), 0) \equiv \mathsf{ack}(m, 1), \tag{15}$$

$$\mathsf{ack}(\mathsf{succ}(m), \mathsf{succ}(n)) \equiv \mathsf{ack}(m, \mathsf{ack}(\mathsf{succ}(m), n)). \tag{16}$$

Solution. I have used John Dougherty's text

https://github.com/jdoughertyii/hott-exercises

We try to solve Equation

$$\mathsf{ack} :\equiv \mathsf{rec}_{\mathbb{N}}(C, c_0, c_s) \tag{17}$$

for

$$C : \mathcal{U}, \quad c_0 : C, \quad c_s : \mathbb{N} \to C \to C.$$

We get $C :\equiv (\mathbb{N} \to \mathbb{N})$. In view of (14), evaluation of (17) at 0 gives $c_0 \equiv \mathsf{succ}$, so that we get

$$\mathsf{ack} :\equiv \mathsf{rec}_{\mathbb{N}}(\mathbb{N} \to \mathbb{N}, \mathsf{succ}, c_s). \tag{18}$$

Let $i : \mathbb{N}, f : \mathbb{N} \to \mathbb{N}$. It suffices to solve Equation

$$c_s(i, f) :\equiv \mathsf{rec}_{\mathbb{N}}(D(i, f), d_0(i, f), d_s(i, f)) \tag{19}$$

for

$$D(i, f) : \mathcal{U}, \quad d_0(i, f) : D(i, f), \quad d_s(i, f) : \mathbb{N} \to D(i, f) \to D(i, f).$$

As we have $c_s(i, f) : \mathbb{N} \to \mathbb{N}$ and $\mathsf{rec}_{\mathbb{N}}(D(i, f), d_0(i, f), d_s(i, f)) : \mathbb{N} \to D(i, f)$, we set $D(i, f) :\equiv \mathbb{N}$, so that we get

$$c_s(i, f) :\equiv \mathsf{rec}_{\mathbb{N}}(\mathbb{N}, d_0(i, f), d_s(i, f)) \tag{20}$$

with

$$d_0(i, f) : \mathbb{N}, \quad d_s(i, f) : \mathbb{N} \to \mathbb{N} \to \mathbb{N}.$$

We have

$$\begin{aligned} &\mathsf{ack}(\mathsf{succ}(m)) \\ &\overset{\text{(a)}}{\equiv} \mathsf{rec}_{\mathbb{N}}(\mathbb{N} \to \mathbb{N}, \mathsf{succ}, c_s, \mathsf{succ}(m)) \\ &\equiv c_s(m, \mathsf{rec}_{\mathbb{N}}(\mathbb{N} \to \mathbb{N}, \mathsf{succ}, c_s, m)) \end{aligned}$$

$$\begin{aligned} &\overset{(b)}{\equiv} c_s(m, \mathsf{ack}(m)) \\ &\overset{(c)}{\equiv} \mathsf{rec}_{\mathbb{N}}(\mathbb{N}, d_0(m, \mathsf{ack}(m)), d_s(m, \mathsf{ack}(m))), \end{aligned}$$

where (a) and (b) follow from (18), and (c) follows from (20). In view of (15), evaluating

$$\mathsf{ack}(\mathsf{succ}(m)) \equiv \mathsf{rec}_{\mathbb{N}}(\mathbb{N}, d_0(m, \mathsf{ack}(m)), d_s(m, \mathsf{ack}(m))) \tag{21}$$

at 0 gives $d_0(m, \mathsf{ack}(m)) \equiv \mathsf{ack}(m)(1)$, prompting us to set $d_0(i, f) :\equiv f(1)$, so that (20) becomes

$$c_s(i, f) :\equiv \mathsf{rec}_{\mathbb{N}}(\mathbb{N}, f(1), d_s(i, f)) \tag{22}$$

and (21) becomes

$$\mathsf{ack}(\mathsf{succ}(m)) \equiv \mathsf{rec}_{\mathbb{N}}\Big(\mathbb{N}, \mathsf{ack}(m, 1), d_s\big(m, \mathsf{ack}(m)\big)\Big). \tag{23}$$

We have

$$\begin{aligned} &\mathsf{ack}(m)\big(\mathsf{ack}(\mathsf{succ}(m), n)\big) \\ &\overset{(a)}{\equiv} \mathsf{ack}(\mathsf{succ}(m), \mathsf{succ}(n)) \\ &\overset{(b)}{\equiv} \mathsf{rec}_{\mathbb{N}}\big(\mathbb{N}, \mathsf{ack}(m, 1), d_s(m, \mathsf{ack}(m)), \mathsf{succ}(n)\big) \\ &\equiv d_s\big(m, \mathsf{ack}(m)\big)\Big(n, \mathsf{rec}_{\mathbb{N}}\Big(\mathbb{N}, \mathsf{ack}(m, 1), d_s\big(m, \mathsf{ack}(m)\big), n\Big)\Big) \\ &\overset{(c)}{\equiv} d_s\big(m, \mathsf{ack}(m)\big)\Big(n, \mathsf{ack}\big(\mathsf{succ}(m), n\big)\Big), \end{aligned}$$

where (a) follows from (16), and (b) and (c) follow from (23). This prompts us to set $d_s(i, f)(x, y) :\equiv f(y)$. Abbreviating $d_s(i, f)$ by f', we get

$$\mathsf{ack} :\equiv \mathsf{rec}_{\mathbb{N}}(\mathbb{N} \to \mathbb{N}, \mathsf{succ}, c_s)$$

with

$$c_s(i, f) \equiv \mathsf{rec}_{\mathbb{N}}(\mathbb{N}, f(1), f')$$

and

$$i : \mathbb{N}, \quad f : \mathbb{N} \to \mathbb{N}, \quad f' : \mathbb{N} \to \mathbb{N} \to \mathbb{N}, \quad f'(x, y) :\equiv f(y),$$

that is

$$\mathsf{ack} :\equiv \mathsf{rec}_{\mathbb{N}}\Big(\mathbb{N} \to \mathbb{N}, \mathsf{succ}, \lambda i.\lambda f.\mathsf{rec}_{\mathbb{N}}\big(\mathbb{N}, f(1), \lambda x.\lambda y.f(y)\big)\Big). \ \square$$

1.7 Exercise 1.13

Statement. Using propositions-as-types, derive the double negation of the principle of excluded middle, i.e. prove *not (not (P or not P)).*

Solution. For any types A, B we define $f : \Big(\big(A + (A \to B)\big) \to B\Big) \to B$ by $f(g) :\equiv (g \circ \mathsf{inr})(g \circ \mathsf{inl})$. Putting $B :\equiv \mathbf{0}$ gives the desired result. □

2 Chapter 2

2.1 Theorem 2.7.2

Recall that $P : A \to \mathcal{U}$ is a type family over a type A, and that $w, w' : \sum_{a:A} P(a)$. Theorem 2.7.2 can also be stated as follows:

The map

$$(w = w') \to \sum_{p:\mathsf{pr}_1(w)=\mathsf{pr}_1(w')} \big(p_\star(\mathsf{pr}_2(w)) = \mathsf{pr}_2(w')\big)$$

defined

$$r \mapsto (\mathsf{ap}_{\mathsf{pr}_1}(r), \mathsf{apd}_{\mathsf{pr}_2}(r))$$

admits an inverse

$$\mathsf{pair}^= : \left(\sum_{p:\mathsf{pr}_1(w)=\mathsf{pr}_1(w')} \big(p_\star(\mathsf{pr}_2(w)) = \mathsf{pr}_2(w')\big)\right) \to (w = w'). \tag{24}$$

We often write $\mathsf{pair}^=_P$ for $\mathsf{pair}^=$ to stress the dependence on the type family $P : A \to \mathcal{U}$.

2.2 Comment after Corollary 2.7.3

In the above context, we have the propositional computation rules

$$\mathsf{ap}_{\mathsf{pr}_1}(\mathsf{pair}^=(p, q)) = p,$$

$$\mathsf{apd}_{\mathsf{pr}_2}(\mathsf{pair}^=(p, q)) = q$$

for $(p,q) : \sum_{p:\mathsf{pr}_1(w)=\mathsf{pr}_1(w')}(p_\star(\mathsf{pr}_2(w)) = \mathsf{pr}_2(w'))$, and the propositional uniqueness principle

$$r = \mathsf{pair}^=(\mathsf{ap}_{\mathsf{pr}_1}(r), \mathsf{apd}_{\mathsf{pr}_2}(r))$$

for $r : w = w'$.

2.3 Theorem 2.7.4

We wish to rewrite the last display of Theorem 2.7.4 in a slightly more explicit way. Recall that we have type families $P : A \to \mathcal{U}$ and $Q : (\sum_{a:A} P(a)) \to \mathcal{U}$. Let us set

$$Q'(a) :\equiv \sum_{u:P(a)} Q(a,u)$$

for $a : A$, let us abbreviate $\mathsf{transport}$ by t, and let us write $\mathsf{pair}^=_P$ instead of $\mathsf{pair}^=$ (see (24) above) to emphasize the fact that this operation is taken with respect to the type family $P : A \to \mathcal{U}$. Recall that $p : x = y$ is a path in A, that $u : P(x)$ and $z : Q(x,u)$, so that $(u,z) : Q'(x)$. Here is the rewriting:

$$\mathsf{t}^{Q'}(p,(u,z)) = \left(\mathsf{t}^P(p,u), \mathsf{t}^Q\left(\mathsf{pair}^=_P\Big(p, \mathsf{refl}_{\mathsf{t}^P(p,u)}\Big), z\right)\right). \tag{25}$$

The key step to check that the above equality is well-typed is the judgment

$$\mathsf{pair}^=_P\Big(p, \mathsf{refl}_{\mathsf{t}^P(p,u)}\Big) : (x,u) =_{\sum_{a:A} P(a)} \big(y, \mathsf{t}^P(p,u)\big).$$

For the reader's convenience, we rewrite (24) more explicitly as

$$\mathsf{pair}^=_P : \left(\sum_{p:\mathsf{pr}_1(w)=\mathsf{pr}_1(w')} \big(\mathsf{t}^P(p, \mathsf{pr}_2(w)) = \mathsf{pr}_2(w')\big)\right) \to (w = w'),$$

so that we have $\mathsf{pair}^=_P(p,q) : w = w'$ if and only if

$$p : \mathsf{pr}_1(w) = \mathsf{pr}_1(w') \quad \text{and} \quad q : \mathsf{t}^P(p, \mathsf{pr}_2(w)) = \mathsf{pr}_2(w').$$

2.4 Commutative diagram summarizing §§ 2.9 and 2.10

Let $f, g : \prod_{a:A} B(a)$, where $B : A \to \mathcal{U}$ is a type family. Recall that

$$(f \sim g) :\equiv \prod_{a:A} (f(a) = g(a)).$$

For $a : A$ we have the commutative diagram

$$\begin{tikzcd}
\boxed{f = g} \arrow[rr, shift left, "\mathsf{happly}"] \arrow[dr, "\mathsf{ev}'_a"'] & & \boxed{f \sim g} \arrow[ll, shift left, "\mathsf{funext}"] \arrow[dl, "\mathsf{ev}_a"] \\
& \boxed{f(a) = g(a)} &
\end{tikzcd}$$

where ev_a is the obvious map (ev stands for *evaluation*), and ev'_a is defined by $\mathsf{ev}'_a :\equiv \mathsf{ev}_a \circ \mathsf{happly}$.

For $A, B : \mathcal{U}$ we have the commutative diagram

$$\begin{tikzcd}
\boxed{A = B} \arrow[rr, shift left, "\mathsf{idtoeqv}"] \arrow[dr, "\mathsf{transport}^{X \mapsto X}"'] & & \boxed{A \simeq B} \arrow[ll, shift left, "\mathsf{ua}"] \arrow[dl, "\mathsf{pr}_1"] \\
& \boxed{A \to B} &
\end{tikzcd}$$

The key point is of course the fact that, in each of the two above diagrams, the horizontal maps are inverses.

2.5 Function Extensionality Axiom (§2.9)

The comments below have been inspired by Lemma 6.3.2 in the HoTT Book.

2.5.1 Part 1

Let $B : A \to \mathcal{U}$ be a type family over a type A, let $f, g : \prod_{a:A} B(a)$ be two dependent functions, and let $h : f \sim g$ be a homotopy from f to g. Define $h' : \prod_{a:A}(I \to B(a))$ by

$$h'(a, 0_I) :\equiv f(a), \quad h'(a, 1_I) :\equiv g(a), \quad h'(a, \mathsf{seg}) := h(a). \tag{26}$$

(I is the interval, with the three constructors $0_I : I,\ 1_I : I,\ \mathsf{seg} : 0_I = 1_I$. See beginning of §6.3 in the book.) Next define $h'' : I \to \prod_{a:A} B(a)$ by

$$h''(i, a) :\equiv h'(a, i).$$

Let $a : A$. We claim

$$\mathsf{happly}(h''(\mathsf{seg}), a) = h'(a, \mathsf{seg}). \tag{27}$$

Let $p : i = j$ be a path in I. It suffices to show $\mathsf{happly}(h''(p), a) = h'(a, p)$, and even, by path induction, $\mathsf{happly}(h''(\mathsf{refl}_i), a) = h'(a, \mathsf{refl}_i)$. But we have

$$\mathsf{happly}(h''(\mathsf{refl}_i), a) = \mathsf{happly}(\mathsf{refl}_f, a) = \mathsf{refl}_{f(a)} = h'(a, \mathsf{refl}_i).$$

This proves (27). Obviously (26) and (27) imply

$$\mathsf{happly}(h''(\mathsf{seg}), a) = h(a). \tag{28}$$

2.5.2 Part 2

We denote the function extensionality axiom by FEA, and we introduce an axiom which we call FEA' and which is clearly implied by FEA, and we'll prove in Part 3 that FEA' implies FEA. FEA' says that for all loop $\ell : f = f$ we have

$$\mathsf{FEA}' : \left(\prod_{a:A} (\mathsf{happly}(\ell, a) = \mathsf{refl}_{f(a)}) \right) \to (\ell = \mathsf{refl}_f). \tag{29}$$

In Part 2 we admit FEA (and thus FEA'). Let $h : f \sim g$. We claim

$$\boxed{\mathsf{funext}(h) = h''(\mathsf{seg})} \tag{30}$$

Let $a : A$. By FEA and path induction we can assume

$$h(a) \equiv \mathsf{refl}_{f(a)}. \tag{31}$$

By FEA again, it suffices to show

$$\mathsf{refl}_{f(a)} = \mathsf{happly}(h''(\mathsf{seg}), a).$$

But this follows from (31) and (28). This proves (30).

2.5.3 Part 3

Recall that FEA stands for "function extensionality axiom" and FEA' is defined in (29). We prove

$$\boxed{\mathsf{FEA}' \text{ implies } \mathsf{FEA}} \tag{32}$$

as follows. We assume FEA'. Let h, h', h'' be as in Part 1, and set

$$\mathsf{funext}(h) :\equiv h''(\mathsf{seg}). \tag{33}$$

We claim

$$\mathsf{funext} \circ \mathsf{happly} = \mathsf{id}_{f=g}. \tag{34}$$

Let

$$h :\equiv \mathsf{happly}(\mathsf{refl}_f). \tag{35}$$

By path induction and (33) it suffices to show $h''(\mathsf{seg}) = \mathsf{refl}_f$. Let $a : A$. By FEA' it even suffices to show $\mathsf{happly}(h''(\mathsf{seg}), a) = \mathsf{refl}_{f(a)}$. But this follows from (28) and (35). This proves (34). We claim

$$\mathsf{happly} \circ \mathsf{funext} = \mathsf{id}_{f\sim g}. \tag{36}$$

Let $h : f \sim g$. We must show $\mathsf{happly}(\mathsf{funext}(h)) = h$. Let $a : A$. By FEA' it suffices to show $\mathsf{happly}(\mathsf{funext}(h), a) = h(a)$. But this follows from (33) and (28). This proves (36). Now (34) and (36) imply (32).

2.6 Definition of the map happly in (2.9.2)

The map

$$\mathsf{happly} : (f = g) \to \prod_{a:A} \big(f(a) = g(a)\big)$$

for $f, g : \prod_{x:A} B(x)$ is defined in the obvious way by path induction. If

$$e : \prod_{x:A} \left(\prod_{y:A} B(y)\right) \to B(x)$$

is the evaluation, *i.e.* $e(x)(f) :\equiv f(x)$, then we have, again by path induction,

$$\mathsf{happly}(p)(x) = \mathsf{ap}_{e(x)}(p)$$

for $p : f = g$ and $x : A$.

2.7 Display preceding (2.9.5)

Note that $(p^{-1})_\star(a)$ and $\mathsf{transport}^A(p^{-1}, a)$ designate the same object, and that we have

$$\mathsf{pair}^=(p^{-1}, \mathsf{refl}_{(p^{-1})_\star(a)}) : (x_2, a) =_{\sum_{x:X} A(x)} \big(x_1, (p^{-1})_\star(a)\big).$$

2.8 Lemma 2.9.7

In this Section we abbreviate $\mathsf{transport}$ by t. Let

$$A : X \to \mathcal{U}, \quad B : \prod_{x:X} A(x) \to \mathcal{U}, \quad \widehat{B} : \left(\sum_{x:X} A(x)\right) \to \mathcal{U}, \quad B' : X \to \mathcal{U}$$

be type families such that $\widehat{B}(w) :\equiv B(\mathsf{pr}_1(w), \mathsf{pr}_2(w))$ for all $w : \sum_{x:X} A(x)$, and

$$B'(x) :\equiv \prod_{a:A(x)} B(x,a)$$

for all $x : X$; let

$$p : x =_X y, \quad f : B'(x), \quad g : B'(y), \quad p_\star :\equiv \mathsf{t}^A(p);$$

and define

$$h : \prod_{a:A(x)} B(y, p_\star a)$$

by setting

$$h(a) :\equiv \mathsf{t}^{\widehat{B}}\Big(\mathsf{pair}^=_A\big(p, \mathsf{refl}_{p_\star a}\big), f(a)\Big) \tag{37}$$

for all $a : A(x)$.

Here are more details about (37): We write $\mathsf{pair}^=_A$ for $\mathsf{pair}^=$ to stress the fact that the pairing is taken with respect to the type family A. Recall that $\mathsf{pair}^=$ is defined in (24) p. 14, and note that we have

$$\mathsf{pair}^=_A\big(p, \mathsf{refl}_{p_\star a}\big) : (x,a) =_{\sum_{x:X} A(x)} (y, p_\star a).$$

Claim: *The maps*

$$\Big(\mathsf{t}^{B'}(p,f) \sim g\Big) \underset{v}{\overset{u}{\rightleftarrows}} \Big(h \sim g \circ p_\star\Big) \tag{38}$$

defined below are inverses.

Proof. We define $k : \mathsf{t}^{B'}(p,f) \circ p_\star \sim h$ by induction on the path $p : x = y$. For $a' : A(y)$ let

$$r(a') : \mathsf{t}^{B'}(p,f)(a') = \mathsf{t}^{B'}(p,f)\Big(p_\star\ (p^{-1})_\star\ a'\Big)$$

and

$$s(a') : g\Big(p_\star\ (p^{-1})_\star\ a'\Big) = g(a')$$

be the obvious paths. We now define u and v in (38) as follows. For $\alpha : \mathsf{t}^{B'}(p, f) \sim g$ and $a : A(x)$ we define $u(\alpha)(a) : h(a) = g(p_\star a)$ by

$$u(\alpha)(a) :\equiv k(a)^{-1} \bullet \alpha(p_\star a),$$

and for $\beta : h \sim g \circ p_\star$ we define $v(\beta) : \mathsf{t}^{B'}(p, f) \sim g$ by

$$v(\beta)(a') :\equiv r(a') \bullet k\big((p^{-1})_\star\, a'\big) \bullet \beta\big((p^{-1})_\star\, a'\big) \bullet s(a').$$

The claim is easily proved by induction on the path $p : x = y$. □

The path

$$r(a') \bullet k\big((p^{-1})_\star\, a'\big) \bullet \beta\big((p^{-1})_\star\, a'\big) \bullet s(a') : \mathsf{t}^{B'}(p, f)(a') = g(a')$$

above is the composite of the paths

$$\begin{aligned}
\mathsf{t}^{B'}(p, f)(a') &\overset{r(a')}{=\!=\!=} \mathsf{t}^{B'}(p, f)\Big(p_\star\, (p^{-1})_\star\, a'\Big) \\
&\overset{k((p^{-1})_\star a')}{=\!=\!=\!=\!=\!=\!=} h\Big((p^{-1})_\star\, a'\Big) \\
&\overset{\beta((p^{-1})_\star a')}{=\!=\!=\!=\!=\!=\!=} g\Big(p_\star\, (p^{-1})_\star\, a'\Big) \\
&\overset{s(a')}{=\!=\!=} g(a').
\end{aligned}$$

2.9 Proof of Theorem 2.11.1

Let $f : A \to B$ be invertible, and let $a, a' : A$. We must show that

$$\mathsf{ap}_f : (a = a') \to (fa = fa')$$

is invertible.

Step 1. (This step is the the same as in the book; I spell it out for the reader's convenience.) Let $g : B \to A$ be an inverse to f, let $\beta : g \circ f = \mathsf{id}_A$, and define

$$\varphi : (gfa = gfa') \to (a = a') \quad \text{by} \quad \varphi(q) :\equiv \beta_a^{-1} \bullet q \bullet \beta_{a'}.$$

Note, as in the book, that the equality $\varphi \circ \mathsf{ap}_g \circ \mathsf{ap}_f = \mathsf{id}_{a=a'}$ follows from the functoriality of ap and the naturality of homotopies, Lemmas 2.2.2 and 2.4.3. (Here $\mathsf{ap}_g : (fa = fa') \to (gfa = gfa')$.)

Step 2. Since φ is invertible (see Example 2.4.8 in the book), $\mathsf{ap}_g \circ \mathsf{ap}_f$ is also invertible, and so is, by symmetry, $\mathsf{ap}_f \circ \mathsf{ap}_g$. Exercise 4.5 (see §4.10 p. 41) implies that ap_f and ap_g are invertible. □

2.10 Proof of Theorem 2.12.5

The equality

$$\mathsf{encode}(\mathsf{inl}(a), \mathsf{decode}(\mathsf{inl}(a), c)) = c$$

for $c : a_0 = a$ can be proved by based path induction. □

2.11 Double induction in §2.13

Claim. *For any type family* $C : \mathbb{N} \to \mathbb{N} \to \mathcal{U}$*, any dependent functions*

$$g : \prod_{n:N} C(0, n), \quad h : \prod_{m:N} C(\mathsf{succ}(m), 0),$$

$$k : \prod_{m,n:N} C(m, n) \to C(\mathsf{succ}(m), \mathsf{succ}(n)),$$

there is a dependent function $f : \prod_{m,n:\mathbb{N}} C(m, n)$ *satisfying*

$$f(0) \equiv g, \quad f(\mathsf{succ}(m), 0) \equiv h(m), \quad f(\mathsf{succ}(m), \mathsf{succ}(n)) \equiv k(m, n, f(m, n))$$

for all $m, n : \mathbb{N}$.

Proof. Define $C' : \mathbb{N} \to \mathcal{U}$ by

$$C'(m) :\equiv \prod_{n:\mathbb{N}} C(m, n).$$

Let $m : \mathbb{N}$ and $x : C'(m)$, and define $c_s(m, x) : C'(\mathsf{succ}(m))$ by

$$c_s(m, x)(0) :\equiv h(m), \quad c_s(m, x)(\mathsf{succ}(n)) :\equiv k(m, n, x(n)).$$

As

$$c_s : \prod_{m:\mathbb{N}} C'(m) \to C'(\mathsf{succ}(m)),$$

we can define $f : \prod_{m:\mathbb{N}} C'(m)$ by

$$f(0) :\equiv g, \quad f(\mathsf{succ}(m)) :\equiv c_s(m, f(m)).$$

We leave the end of the proof to the reader. □

2.12 Section 2.14.1

2.12.1 Use of Theorem 2.7.4

We apply Theorem 2.7.4 (see §2.3 p. 15) to §2.14.1 of the book by defining $P : \mathcal{U} \to \mathcal{U}$ by $P(A) :\equiv (A \to A \to A)$, by defining

$$Q : \left(\sum_{A:\mathcal{U}} P(A)\right) \to \mathcal{U}$$

by

$$Q(A, m) :\equiv \prod_{x,y,z:A} \big(m(x, m(y, z)) = m(m(x, y), z)\big),$$

and by defining $Q' : \mathcal{U} \to \mathcal{U}$ by

$$Q'(A) :\equiv \sum_{m:P(A)} Q(A, m).$$

In the notation of §2.3 p. 15 the last display of Theorem 2.7.4 (see (25) p. 15) now reads

$$\mathsf{t}^{Q'}(p, (m, a)) = \left(\mathsf{t}^P(p, m), \mathsf{t}^Q\left(\mathsf{pair}^{=}_{\overline{P}}\left(p, \mathsf{refl}_{\mathsf{t}^P(p,m)}\right), a\right)\right)$$

with $p :\equiv \mathsf{ua}(e)$. □

2.12.2 A general comment

The main purpose of §2.14.1 is to obtain the formulas

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

and

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

(We refer the reader to the book for the precise notation. The above chain of equalities is preceded by the sentence "Moreover, though we do not show the proof, one can calculate that the induced proof that m' is associative $\dots$ is equal to a function sending $b_1, b_2, b_3 : B$ to a path given by the following steps". The main ingredient is Theorem 2.7.4 — see §2.3 p. 15.)

It seems to me one could argue as follows: state the above two formulas, check that they are well-typed, check that they hold when $A \equiv B$ and p is reflexivity, conclude by path induction that they hold in general.

2.13 Exercise 2.7

Statement. State and prove a generalization of Theorem 2.6.5 from cartesian products to Σ-types.

Solution. We shall freely use the proof of Theorem 2.7.2 (see §2.1 p. 14). Let $B : A \to \mathcal{U}$, and let us denote respectively by α and β the dependent functions f and g in the proof of Theorem 2.7.2:

$$\prod_{z,w:\sum_{a:A} B(a)} \left((z = w) \overset{\alpha}{\underset{\beta}{\rightleftarrows}} \sum_{p:\mathsf{pr}_1(z)=\mathsf{pr}_1(w)} p_\star \mathsf{pr}_2(z) = \mathsf{pr}_2(w) \right).$$

Note that the expression $\beta(z, w)$ has the same meaning as the expression $\mathsf{pair}^=$ introduced in the book just before Theorem 2.7.4. Let $B' : A' \to \mathcal{U}$ and set similarly

$$\prod_{z',w':\sum_{a':A'} B'(a')} \left((z' = w') \overset{\alpha'}{\underset{\beta'}{\rightleftarrows}} \sum_{p':\mathsf{pr}_1(z')=\mathsf{pr}_1(w')} p'_\star \mathsf{pr}_2(z') = \mathsf{pr}_2(w') \right).$$

Let $g : A \to A'$, $h : \prod_{a:A} B(a) \to B'(g(a))$. Cleary, there is a map

$$f : \left(\sum_{a:A} B(a) \right) \to \sum_{a':A'} B'(a')$$

satisfying $f(a, b) \equiv (g(a), h(a, b))$ for all $a : A$ and all $b : B(a)$.

Let

$$z, w : \sum_{a:A} B(a); \quad p : \mathsf{pr}_1(z) = \mathsf{pr}_1(w); \quad q : p_\star \mathsf{pr}_2(z) = \mathsf{pr}_2(w).$$

We claim

$$\mathsf{ap}_f\big(\beta(z,w,p,q)\big) = \beta'\big(f(z), f(w), \mathsf{ap}_g(p), \mathsf{apd}_h(q)\big). \tag{39}$$

Proof of (39). For z and w as above, and $r : z = w$ set

$$p(r) :\equiv \mathsf{pr}_1(\alpha(z,w,r)), \quad q(r) :\equiv \mathsf{pr}_2(\alpha(z,w,r)).$$

Then we can prove

$$\mathsf{ap}_f\big(\beta(z,w,p(r),q(r))\big) = \beta'\big(f(z), f(w), \mathsf{ap}_g(p(r)), \mathsf{apd}_h(q(r))\big) \tag{40}$$

by induction on r. Clearly (40) implies (39). □

2.14 Exercise 2.10

The result of Exercise 2.10 can be stated as follows:

If $B : A \to \mathcal{U}$ and $C : (\sum_{a:A} B(a)) \to \mathcal{U}$ are type families, then the maps

$$\begin{tikzcd}
\left(\sum_{a:A}\sum_{b:B(a)} C((a,b))\right) \arrow[r, shift left, "f"] & \left(\sum_{z:\sum_{a:A} B(a)} C(z)\right) \arrow[l, shift left, "g"]
\end{tikzcd}$$

defined by

$$f(a,b,c) :\equiv ((a,b),c), \quad g(z,c) :\equiv (\mathsf{pr}_1 z, \mathsf{pr}_2 z, c)$$

are inverses.

The proof is straightforward.

2.15 Exercise 2.14

Statement. Suppose we add to type theory the *equality reflection rule* which says that if there is an element $p : x = y$, then in fact $x \equiv y$. Prove that for any $p : x = x$ we have $p \equiv \mathsf{refl}_x$. (This implies that every type is a *set* in the sense to be introduced in §3.1; see §7.2.)

Solution. Let $p : x = y$. We have $p : x = x$ by the equality reflection rule, and $p = \mathsf{refl}_x$ by path induction. □

3 Chapter 3

3.1 Proof of Lemma 3.1.8

Recall the statement:

If A is a set (that is, $\mathsf{isSet}(A)$ is inhabited), then A is a 1-type.

Here is a slightly different wording of the proof:

Suppose $f : \mathsf{isSet}(A)$ and let $x, y : A$. For any $p, q : x = y$ we have $f(p, q) : p = q$. For $q' : x = y$ and $r : q = q'$ we prove

$$r = f(p, q)^{-1} \cdot f(p, q')$$

by path induction on r. □

3.2 Theorem 3.2.2

Recall the statement:

It is not the case that for all $A : \mathcal{U}$ we have $\neg\neg A \to A$.

Here is a minor variant. Set

$$F :\equiv \prod_{A:\mathcal{U}} \neg\neg A \to A, \qquad G :\equiv \prod_{A:\mathcal{U}} \|\mathbf{2} = A\| \to A.$$

It suffices to prove $F \to G$ and $\neg G$.

Proof of $F \to G$. Given $f : F$ we must define $g : G$. It suffices define

$$h : \|\mathbf{2} = A\| \to \neg\neg A$$

for $A : \mathcal{U}$, and to set $g(A, u) :\equiv f(A, h(u))$. It suffices in turn to define

$$h' : (\mathbf{2} = A) \to \neg\neg A.$$

To do this, we set $h'(p, k) :\equiv k(\mathsf{transport}^{X \mapsto X}(p, 0_{\mathbf{2}}))$ for all $k : \neg A$.

Proof of $\neg G$. Similar to the proof of $\neg F$ (that is, the proof of Theorem 3.2.2) in the book. □

The (limited) interest of this comment is that, given a type B, it might be easier to prove $B \to G$ than to prove $B \to F$ (or to prove $B \to \mathbf{0}$ directly).

3.3 Proof of Lemma 3.3.4

Recall the statement of Lemma 3.3.4: *Every mere proposition is a set.* In the proof of Lemma 3.1.8 in §3.1 above we derived the conclusion that $x = y$ is a set using only the assumption that $x = y$ is a mere proposition.

3.4 Contractibility (§3.11)

For the reader's convenience we paste below most of the statements in §3.11 of the book.

Definition 3.11.1. A type A is **contractible**, or a **singleton**, if there is $a : A$, called the **center of contraction**, such that $a = x$ for all $x : A$. We denote the specified path $a = x$ by contr_x.

In other words, the type $\mathsf{isContr}(A)$ is defined to be

$$\mathsf{isContr}(A) :\equiv \sum_{a:A} \prod_{x:A} (a = x).$$

Lemma 3.11.3. *For a type A, the following are logically equivalent*

1. *A is contractible in the sense of Definition 3.11.1.*
2. *A is a mere proposition, and there is a point $a : A$.*
3. *$A = \mathbf{1}$.*

Lemma 3.11.4. *For any type A, the type $\mathsf{isContr}(A)$ is a mere proposition.*

Corollary 3.11.5. *If A is contractible, then so is $\mathsf{isContr}(A)$.*

Lemma 3.11.6. *If $P : A \to \mathcal{U}$ is a type family such that each $P(a)$ is contractible, then $\prod_{x:A} P(x)$ is contractible.*

Of course, if A is equivalent to B and A is contractible, then so is B. More generally, it suffices for B to be a *retract* of A. By definition, a **retraction** is a function $r : A \to B$ such that there exists a function $s : B \to A$, called its **section**, and a homotopy $\varepsilon : \prod_{y:B}(r(s(y)) = y)$; then we say that B is a **retract** of A.

Lemma 3.11.7. *If B is a retract of A, and A is contractible, then so is B.*

Lemma 3.11.8. *For any A and any $a : A$, the type $\sum_{x:A}(a = x)$ is contractible.*

When this happens, it can allow us to simplify a complicated construction up to equivalence, using the informal principle that contractible data can be freely ignored. This principle consists of many lemmas, most of which we leave to the reader; the following is an example.

Lemma 3.11.9. *Let $P : A \to \mathcal{U}$ be a type family.*

1. *If each $P(x)$ is contractible, then $\sum_{x:A} P(x)$ is equivalent to A.*
2. *If A is contractible and $a : A$, then $\sum_{x:A} P(x)$ is equivalent to $P(a)$.*

Lemma 3.11.10. *A type A is a mere proposition if and only if for all $x, y : A$, the type $x =_A y$ is contractible.*

3.5 Proof of Lemma 3.11.4

Recall the statement (see §3.4): *For any type A, the type* $\mathsf{isContr}(A)$ *is a mere proposition.*

Here is a minor variant of the proof:

Let $c, c' : \mathsf{isContr}(A)$. Since A is contractible (by c or c'), Lemma 3.11.3 (see §3.4 above) implies $A = \mathbf{1}$, and thus

$$\mathsf{isContr}(A) = \left(\sum_{a:\mathbf{1}} \prod_{x:\mathbf{1}} (a = x)\right) = \left(\sum_{a:\mathbf{1}} \prod_{x:\mathbf{1}} \mathbf{1}\right) = \left(\sum_{a:\mathbf{1}} \mathbf{1}\right) = \mathbf{1}. \square$$

3.6 Proof of Lemma 3.11.8

Recall the statement (see §3.4):

For any A and any $a : A$, the type $\sum_{x:A}(a = x)$ is contractible.

Here is a minor variant of the proof:

We prove $(x, p) = (a, \mathsf{refl}_a)$ for any $(x, p) : \sum_{x:A}(a = x)$ by based path induction. □

3.7 Exercise 3.5

Statement. Show that $\mathsf{isProp}(A) \simeq (A \to \mathsf{isContr}(A))$.

Solution. Recall the definitions:

$$\mathsf{isProp}(A) :\equiv \prod_{a,b:A} (a = b), \quad \mathsf{isContr}(A) :\equiv \sum_{a:A} \prod_{b:A} (a = b).$$

By Lemma 3.3.5 the type $\mathsf{isProp}(A)$ is a mere proposition. By Example 3.6.2 and Lemma 3.11.4 (see §3.4 p. 26) the type $A \to \mathsf{isContr}(A)$ is also a mere proposition. By Lemma 3.3.3 it suffices to show that $\mathsf{isProp}(A)$ and $A \to \mathsf{isContr}(A)$ are logically equivalent.

We define

$$u : \mathsf{isProp}(A) \to (A \to \mathsf{isContr}(A)), \quad v : (A \to \mathsf{isContr}(A)) \to \mathsf{isProp}(A)$$

by

$$u(f, a) :\equiv (a, f(a)), \quad v(g)(a, b) :\equiv \mathsf{pr}_2(g(a))(a)^{-1} \bullet \mathsf{pr}_2(g(a))(b).\ \Box$$

3.8 Exercise 3.6

Statement. Show that if A is a mere proposition, then so is $A + \neg A$.

Proof. For $a_1, a_2 : A$ we have $a_1 = a_2$ because A is a mere proposition, and thus $\mathsf{inl}(a_1) = \mathsf{inl}(a_2)$. For $a_1', a_2' : \neg A$ we have $a_1' = a_2'$ because $\neg B$ is a mere proposition for any type B, and thus $\mathsf{inr}(a_1') = \mathsf{inr}(a_2')$. Let $a : A$ and $a' : \neg A$. We must show $\mathsf{inl}(a) = \mathsf{inr}(a')$. Let $f : \mathbf{0} \to (\mathsf{inl}(a) = \mathsf{inr}(a'))$, and note that $f(a'(a)) : \mathsf{inl}(a) = \mathsf{inr}(a')$. □

3.9 Exercise 3.7

Statement. Show that if A and B are mere propositions and $\neg(A \times B)$, then $A+B$ is also a mere proposition.

Proof. It suffices to prove $x = y$ in the following cases:

(a) $x = \mathsf{inl}(a),\ y = \mathsf{inl}(a')$,

(b) $x = \mathsf{inr}(b),\ y = \mathsf{inr}(b')$,

(c) $x = \mathsf{inl}(a),\ y = \mathsf{inr}(b)$.

We leave Cases (a) and (b) to the reader, and take up Case (c). Let $f : (A \times B) \to \mathbf{0}$ and $g : \mathbf{0} \to (\mathsf{inl}(a) = \mathsf{inr}(b))$. We get $g(f(a, b)) : \mathsf{inl}(a) = \mathsf{inr}(b)$. □

3.10 Exercise 3.9

We admit the Law of Excluded Middle LEM defined in (3.4.1) by

$$\mathsf{LEM} : \prod_{A:\mathcal{U}} \mathsf{isProp}(A) \to (A + \neg A),$$

and we want to prove

$$\left(\sum_{A:\mathcal{U}} \mathsf{isProp}(A)\right) \simeq \mathbf{2}. \tag{41}$$

Proof of (41). By Lemma 3.3.2 we have

$$f_1 : \prod_{A:\mathcal{U}} \mathsf{isProp}(A) \to A \to (A = \mathbf{1}).$$

It is easy to see that we have

$$f_2 : \prod_{A:\mathcal{U}} \neg A \to (A = \mathbf{0}).$$

Define

$$f_3 : \prod_{A:\mathcal{U}} \mathsf{isProp}(A) \to (A + \neg A) \to ((A = \mathbf{1}) + (A = \mathbf{0}))$$

by

$$f_3(A, p, \mathsf{inl}(a)) :\equiv \mathsf{inl}(f_1(A, p, a)), \qquad f_3(A, p, \mathsf{inr}(a')) :\equiv \mathsf{inr}(f_2(A, a')).$$

Define

$$f_4 : \prod_{A:\mathcal{U}} \mathsf{isProp}(A) \to \big((A = \mathbf{1}) + (A = \mathbf{0})\big)$$

by

$$f_4(A, p) :\equiv f_3(A, p, \mathsf{LEM}(A, p)).$$

In view of f_4 we can define

$$f_5 : \left(\sum_{A:\mathcal{U}} \mathsf{isProp}(A)\right) \to \mathbf{2}$$

by

$$f_5(A, p) :\equiv \begin{cases} 1_{\mathbf{2}} & \text{if } A = \mathbf{1} \\ 0_{\mathbf{2}} & \text{if } A = \mathbf{0}. \end{cases}$$

Let $p_0 : \mathsf{isProp}(\mathbf{0})$ and $p_1 : \mathsf{isProp}(\mathbf{1})$, and define

$$f_6 : \mathbf{2} \to \sum_{A:\mathcal{U}} \mathsf{isProp}(A)$$

by $f_6(0_{\mathbf{2}}) :\equiv (\mathbf{0}, p_0)$ and $f_6(1_{\mathbf{2}}) :\equiv (\mathbf{1}, p_1)$.

It is straightforward to check that $f_5 \circ f_6 = \mathsf{id}_{\mathbf{2}}$.

Let $A : \mathcal{U}$ and $p : \mathsf{isProp}(A)$. It suffices to show

$$f_6(f_5(A, p)) = (A, p). \tag{42}$$

In view of f_4 it suffices to prove that (42) holds if $A = \mathbf{1}$ or if $A = \mathbf{0}$, which is easy. □

Note that the Law of Excluded Middle LEM is equivalent to

$$\mathsf{LEM}' : \prod_{A:\mathcal{U}} \mathsf{isProp}(A) \to \big((A = \mathbf{1}) + (A = \mathbf{0})\big).$$

3.11 Exercise 3.17

Here is a slightly more precise statement:

Show that the rules for the propositional truncation given in §3.7 of the book are sufficient to imply the following induction principle: given $B : \|A\| \to \mathsf{Prop}$ *and* $f : \prod_{a:A} B(|a|)$, *there is a* $g : \prod_{x:\|A\|} B(x)$ *such that* $g(|a|) = f(a)$ *for all* $a : A$.

Proof. Let $p : \mathsf{isProp}(\|A\|)$ (*i.e.* $p(x, y) : x = y$ for all $x, y : \|A\|$), and let $x : \|A\|$. The map $h : A \to B(x)$ defined by

$$h(a) :\equiv p(|a|, x)_\star(f(a))$$

induces a map $k : \|A\| \to B(x)$ such that $k(|a|) \equiv h(a)$ for all $a : A$, and we can set $g(x) :\equiv k(x)$ for all $x : \|A\|$. This yields

$$g(|a|) \equiv k(|a|) \equiv h(a) \equiv p(|a|, |a|)_\star(f(a)) = f(a)$$

for all $a : A$. □

3.12 Exercise 3.18

Statement. Show that the law of excluded middle (3.4.1) and the law of double negation (3.4.2) are logically equivalent.

Recall that the law of excluded middle LEM defined in (3.4.1) by

$$\mathsf{LEM} : \prod_{A:\mathcal{U}} \mathsf{isProp}(A) \to (A + \neg A),$$

and that the law of double negation is defined in (3.4.2) by

$$\prod_{A:\mathcal{U}} \mathsf{isProp}(A) \to (\neg\neg A \to A).$$

Proof of the Statement. To prove that the law of excluded middle implies the law of double negation, we assume that the types

$$\mathsf{isProp}(A), \quad A + \neg A, \quad \neg\neg A$$

are inhabited, and we show that so is A as follows. Let $x : A + \neg A$. If $x \equiv \mathsf{inl}(a)$ for some $a : A$ we are done. If $x \equiv \mathsf{inr}(a')$ for some $a' : \neg A$, we let $f : \mathbf{0} \to A$ and $a'' : \neg\neg A$, and get $f(a''(a')) : A$.

To prove that the law of double negation implies the law of excluded middle, we let A be a mere proposition, and we show that $A + \neg A$ is inhabited as follows. By Exercise 3.6 (see §3.8 p. 28 above), $A+\neg A$ is also a mere proposition, and thus, by double negation, there is a map $\neg\neg(A + \neg A) \to (A + \neg A)$. Hence it suffices to check that $\neg\neg(A + \neg A)$ is inhabited. To this end, we define $x'' : \neg\neg(A + \neg A)$ by $x''(x') :\equiv (x' \circ \mathsf{inr})(x' \circ \mathsf{inl})$. □

4 Chapter 4

4.1 Proof of Lemma 4.1.2

For the reader's convenience we paste the statement and proof of Lemma 4.1.2:

Lemma 4.1.2. *Suppose we have a type A with $a : A$ and $q : a = a$ such that*

(i) The type $a = a$ is a set.

(ii) For all $x : A$ we have $\|a = x\|$.

(iii) For all $p : a = a$ we have $p \bullet q = q \bullet p$.

Then there exists $f : \prod_{x:A}(x = x)$ with $f(a) = q$.

Proof. Let $g : \prod_{x:A}\|a = x\|$ be as given by (ii). First we observe that each type $x =_A y$ is a set. For since being a set is a mere proposition, we may apply the induction principle of propositional truncation, and assume that $g(x) = |p|$ and $g(y) = |p'|$ for $p : a = x$ and $p' : a = y$. In this case, composing with p and p'^{-1} yields an equivalence $(x = y) \simeq (a = a)$. But $a = a$ is a set by (i), so $x = y$ is also a set.

Now, we would like to define f by assigning to each x the path $g(x)^{-1} \bullet q \bullet g(x)$, but this does not work because $g(x)$ does not inhabit $a = x$ but rather $\|a = x\|$, and the type $x = x$ may not be a mere proposition, so we cannot use induction on propositional truncation. Instead we can apply the technique mentioned in §3.9: we characterize uniquely the object we wish to construct. Let us define, for each $x : A$, the type

$$B(x) :\equiv \sum_{r:x=x} \prod_{s:a=x} (r = s^{-1} \bullet q \bullet s).$$

We claim that $B(x)$ is a mere proposition for each $x : A$. Since this claim is itself a mere proposition, we may again apply induction on truncation and assume that $g(x) = |p|$ for some $p : a = x$. Now suppose given (r, h) and (r', h') in $B(x)$; then we have

$$h(p) \bullet h'(p)^{-1} : r = r'.$$

It remains to show that h is identified with h' when transported along this equality, which by transport in identity types and function types (§§2.9 and 2.11), reduces to showing

$$h(s) = h(p) \bullet h'(p)^{-1} \bullet h'(s)$$

for any $s : a = x$. But each side of this is an equality between elements of $x = x$, so it follows from our above observation that $x = x$ is a set.

Thus, each $B(x)$ is a mere proposition; we claim that $\prod_{x:A} B(x)$. Given $x : A$, we may now invoke the induction principle of propositional truncation to assume that $g(x) = |p|$ for $p : a = x$. We define $r :\equiv p^{-1} \bullet q \bullet p$; to inhabit $B(x)$ it remains to show that for any $s : a = x$ we have $r = s^{-1} \bullet q \bullet s$. Manipulating paths, this reduces to showing that $q \bullet (p \bullet s^{-1}) = (p \bullet s^{-1}) \bullet q$. But this is just an instance of (iii). □

Let us denote by $(\star)$ the claim

h is identified with h' when transported along this equality, which by transport in identity types and function types, reduces to showing

$$h(s) = h(p) \bullet h'(p)^{-1} \bullet h'(s)$$

for any $s : a = x$

in the proof of Lemma 4.1.2 (see above). We make three observations:

(1) The fact that $(\star)$ implies that $B(x)$ is a mere proposition follows from Theorem 2.7.2 (see §2.1 p. 14).

(2) The statement $(\star)$ itself follows from the lemma below, which is easily proved by path induction:

Lemma. *Let* A *be a type; let* $a, x : A$*; let* $q : a = a$*; let* $P : (x = x) \to \mathcal{U}$ *be defined by*

$$P(r) :\equiv \prod_{s:a=x} (r = s^{-1} \bullet q \bullet s);$$

and let

$$r, r' : x = x, \qquad t : r = r', \qquad h : P(r), \qquad s : a = x.$$

Then

$$\mathsf{transport}^P(t, h)(s) = t^{-1} \bullet h(s). \ \square$$

(3) Let us spell out the end of the proof. Recall that $B(x)$ is defined by

$$B(x) :\equiv \sum_{r:x=x} \prod_{s:a=x} (r = s^{-1} \bullet q \bullet s).$$

Let $k : \prod_{x:A} B(x)$, and set $f(x) :\equiv \mathsf{pr}_1(k(x))$ for all $x : A$. We must show

$$\text{(a) } f : \prod_{x:A} (x = x) \quad \text{and} \quad \text{(b) } f(a) = q.$$

Claim (a) is clear. Let us prove (b). By assumption (iii) of Lemma 4.1.2 (see above), there is a $u : \prod_{s:a=a}(q = s^{-1} \bullet q \bullet s)$. In particular we have $(q, u) : B(a)$. As $B(a)$ is a mere proposition, this implies $k(a) = (q, u)$, and thus $f(a) = q$. $\square$

4.2 Proof of Lemma 4.2.11

Use also Theorem 2.15.7.

For the reader's convenience we paste the statement of Theorem 2.15.7:

Suppose suppose we have a type X and type families $A : X \to \mathcal{U}$ and $P : \prod_{x:X} A(x) \to \mathcal{U}$. Then we have a function

$$\left(\prod_{x:X} \sum_{a:A(x)} P(x, a) \right) \to \left(\sum_{g:\prod_{x:X} A(x)} \prod_{x:X} P(x, g(x)) \right). \tag{43}$$

defined by $f \mapsto (\mathsf{pr}_1 \circ f, \mathsf{pr}_2 \circ f)$.

Theorem 2.15.7. (43) *is an equivalence.*

4.3 Proof of Lemma 4.7.3

I would replace "applying a version of Lemma 3.11.9" with "using based path induction".

4.4 Proof of Theorem 4.7.6

Let us check Equivalence ($\star$) in the proof of Theorem 4.7.6: Recalling that we are given type families $P, Q : A \to \mathcal{U}$, a dependent function $f : \prod_{a:A} P(a) \to Q(a)$, a path $p : a = x$ in A, and a $v : Q(x)$, set

$$R(a,p) :\equiv \sum_{u:P(a)} \Big(p_\star\big(f(a,u)\big) = v\Big).$$

Claim:

$$\left(\sum_{a:A} \sum_{p:a=x} R(a,p)\right) \overset{\text{(a)}}{\simeq} \left(\sum_{z:\sum_{a:A}(a=x)} R(z)\right) \overset{\text{(b)}}{\simeq} R(x, \mathsf{refl}_x).$$

Proof. Equivalence (a) follows from Exercise 2.10 (see §2.14 p. 24). Equivalence (b) follows from §3.4 p. 26. □

4.5 Proof of Lemma 4.8.1

Recall the statement:

For any type family $B : A \to \mathcal{U}$, the fiber of $\mathsf{pr}_1 : (\sum_{x:A} B(x)) \to A$ over $a : A$ is equivalent to $B(a)$:

$$\mathsf{fib}_{\mathsf{pr}_1}(a) \simeq B(a).$$

Proof. We have

$$\mathsf{fib}_{\mathsf{pr}_1}(a) :\equiv \sum_{u:\sum_{x:A} B(x)} (\mathsf{pr}_1(u) = a)$$

$$\overset{\text{(a)}}{\simeq} \sum_{x:A} \sum_{b:B(x)} (x = a) \simeq \sum_{x:A} \sum_{p:x=a} B(x)$$

$$\overset{(b)}{\simeq} \sum_{z:\sum_{x:A}(x=a)} B(\mathsf{pr}_1(z)) \overset{(c)}{\simeq} B(\mathsf{pr}_1(a,\mathsf{refl}_a)) \equiv B(a),$$

where (a) and (b) follow from Exercise 2.10 (see §2.14 p. 24), and (c) follows from §3.4 p. 26. □

4.6 Proof of Lemma 4.8.2

Here is a minor variant to the proof of Lemma 4.8.2. Recall the statement:

For any function $f : A \to B$, we have $A \simeq \sum_{b:B} \mathsf{fib}_f(b)$.

Recall Definition 4.2.4: *The fiber of a map $f : A \to B$ over a point $b : B$ is*

$$\mathsf{fib}_f(b) :\equiv \sum_{a:A} (f(a) = b).$$

In particular fib_f is a type family over B. The corresponding transport can be described as follows. Let $p : b =_B b'$ and $(a, q) : \mathsf{fib}_f(b)$, that is $a : A$ and $q : f(a) = b$. Then we have

$$p_\star(a, q) = (a, q \bullet p). \tag{44}$$

Set

$$C :\equiv \sum_{b:B} \mathsf{fib}_f(b) :\equiv \sum_{b:B} \sum_{a:A} (f(a) = b).$$

We must show $A \simeq C$. Define $g : A \to C$ and $h : C \to A$ by

$$g(a) :\equiv (f(a), a, \mathsf{refl}_{f(a)}), \qquad h(b, a, p) :\equiv a.$$

We claim that g and h are inverses. It is easy to check $h \circ g = \mathsf{id}_A$. Let $(b, a, p) : C$. It only remains to show $(f(a), a, \mathsf{refl}_{f(a)}) = (b, a, p)$. But this follows from (44). □

4.7 Proof of Theorem 4.8.3

For the reader's convenience we paste the statement and proof of Theorem 4.8.3:

Theorem 4.8.3. *For any type B there is an equivalence*

$$\chi : \left(\sum_{A:\mathcal{U}} (A \to B) \right) \simeq (B \to \mathcal{U}).$$

Proof. We have to construct quasi-inverses

$$\chi : \left(\sum_{A:\mathcal{U}}(A \to B)\right) \to B \to \mathcal{U}$$

$$\psi : (B \to \mathcal{U}) \to \left(\sum_{A:\mathcal{U}}(A \to B)\right).$$

We define χ by $\chi((A, f), b) :\equiv \mathsf{fib}_f(b)$, and ψ by $\psi(P) :\equiv ((\sum_{b:B} P(b)), \mathsf{pr}_1)$.

Now we have to verify that $\chi \circ \psi \simeq \mathsf{id}$ and that $\psi \circ \chi \simeq \mathsf{id}$.

(a) Let $P : B \to \mathcal{U}$. By Lemma 4.8.1 (see §4.5 p. 34), $\mathsf{fib}_{\mathsf{pr}_1}(b) \simeq P(b)$ for any $b : B$, so it follows immediately that $P \simeq \chi(\psi(P))$.

(b) Let $f : A \to B$ be a function. We have to find a path

$$\left(\sum_{b:B} \mathsf{fib}_f(b),\ \mathsf{pr}_1\right) = (A, f).$$

First note that by Lemma 4.8.2 (see §4.6 p. 35), we have $e : \sum_{b:B} \mathsf{fib}_f(b) \simeq A$ with $e(b, a, p) :\equiv a$ and

$$e^{-1}(a) :\equiv (f(a), a, \mathsf{refl}_f(a)).$$

By Theorem 2.7.2 (see §2.1 p. 14), it remains to show $\mathsf{ua}(e)_\star(\mathsf{pr}_1) = f$. But by the computation rule for univalence and (2.9.4) (see (45) below), we have $\mathsf{ua}(e)_\star(\mathsf{pr}_1) = \mathsf{pr}_1 \circ e^{-1}$, and the definition of e^{-1} immediately yields $\mathsf{pr}_1 \circ e^{-1} \equiv f$. □

Here are some details about the last sentence of the above proof:

Firstly we rewrite (2.9.4) as follows. Abbreviating transport by t we have

$$\mathsf{t}^{A\to B}(p, f, x) = \mathsf{t}^B(p, f(\mathsf{t}^A(p^{-1}, x))) \tag{45}$$

for $p : x_1 =_X x_2,\ f : A(x_1) \to B(x_1),\ a_2 : A(x_2)$.

Let the notation of §4.6 above be in force. Let $\mathsf{pr}_1 : C \to B$ be the first projection. We must show

$$(C, \mathsf{pr}_1) =_{\sum_{X:\mathcal{U}} X \to B} (A, f). \tag{46}$$

By §4.6 we have an equivalence $e : C \simeq A$. Set $g :\equiv \mathsf{pr}_1(e^{-1})$ (here of course pr_1 denotes the first projection $(A \simeq C) \to (A \to C)$). Set also $q :\equiv \mathsf{ua}(e)$, so that we get $q : C = A$. We claim

$$q_\star(\mathsf{pr}_1) = f. \tag{47}$$

By Theorem 2.7.2 (see §2.1 p. 14), (47) will imply (46). For all $X : \mathcal{U}$ set $I(X) :\equiv X,\ D(X) :\equiv A$. Let the notation of (45) be in force. In particular, the computation rule for univalence stated right after Remark 2.10.4 in the book reads

$$\mathsf{t}^I(q^{-1}) = g, \tag{48}$$

and (47) becomes

$$\mathsf{t}^{I\to D}(q, \mathsf{pr}_1) = f.$$

Let $a : A$. We must show

$$\mathsf{t}^{I\to D}(q, \mathsf{pr}_1, a) = f(a).$$

We get

$$\mathsf{t}^{I\to D}(q, \mathsf{pr}_1, a) \overset{\text{(a)}}{=} \mathsf{t}^D(q, \mathsf{pr}_1(\mathsf{t}^I(q^{-1}, a))) \overset{\text{(b)}}{=} \mathsf{t}^D(q, \mathsf{pr}_1(g(a))) \overset{\text{(c)}}{=} \mathsf{pr}_1(g(a)) \overset{\text{(d)}}{=} f(a),$$

where (a) follows from (45), (b) follows from (48), (c) follows from Lemma 2.3.5 in the book (see below), and (d) follows from the definition of g. □

Lemma 2.3.5. *If $P : A \to \mathcal{U}$ is defined by $P(x) :\equiv B$ for a fixed $B : \mathcal{U}$, then for any $x, y : A$ and $p : x = y$ and $b : B$ we have a path*

$$\mathsf{transportconst}^B_p(b) : \mathsf{transport}^P(p, b) = b.$$

4.8 Proof of Theorem 4.8.4

Here is the statement:

Let $f : A \to B$ be a function. Then the diagram

$$\begin{tikzcd} A \arrow[r, "\theta_f"] \arrow[d, "f"'] & \mathcal{U}_\bullet \arrow[d, "\mathsf{pr}_1"] \\ B \arrow[r, "\chi(A,f)"'] & \mathcal{U} \end{tikzcd}$$

is a pullback square (see Exercise 2.11). Here the function θ_f is defined by

$$\theta_f(a) :\equiv \Big(\mathsf{fib}_f\big(f(a)\big), \big(a, \mathsf{refl}_{f(a)}\big)\Big).$$

I don't understand the proof of the equivalence

$$A \simeq \sum_{b:B} \sum_{X:\mathcal{U}} \sum_{x:X} (\mathsf{fib}_f(b) = X) \tag{49}$$

at the beginning of the proof of Theorem 4.8.4. Here is how I would present things.

Claim. *For any type A we have*

$$A \simeq \sum_{X:\mathcal{U}} \big(X \times (X = A)\big). \tag{50}$$

Proof. Write t for "transport". Set $C :\equiv \sum_{X:\mathcal{U}} \big(X \times (X = A)\big)$, and define

$$f : A \to C, \qquad g : C \to A$$

by

$$f(a) :\equiv (A, a, \mathsf{refl}_A), \qquad g(B, b, p) :\equiv \mathsf{t}^{X \mapsto X}(p, b).$$

We have $g(f(a)) \equiv a$ for all $a : A$. Let $(B, b, p) : C$. We have

$$f(g(B, b, p)) = (A, \mathsf{t}^{X \mapsto X}(p, b), \mathsf{refl}_A)$$

and we must show

$$(B, b, p) = (A, \mathsf{t}^{X \mapsto X}(p, b), \mathsf{refl}_A).$$

By Theorem 2.7.2 (see §2.1 p. 14) it suffices to verify

$$\mathsf{t}^{X \mapsto X \times (X = A)}(p, (b, p)) = (\mathsf{t}^{X \mapsto X}(p, b), \mathsf{refl}_A).$$

By Theorem 2.6.4 it suffices to prove $\mathsf{t}^{X \mapsto (X = A)}(p, p) = \mathsf{refl}_A$. But this follows from Lemma 2.11.2. □

Going back to the beginning of the proof of Theorem 4.8.4, we get

$$A \simeq \left(\sum_{b:B} \mathsf{fib}_f(b) \right) \simeq \left(\sum_{b:B} \sum_{X:\mathcal{U}} \Big(X \times \big(X = \mathsf{fib}_f(b)\big)\Big) \right)$$

$$\simeq \sum_{b:B} \sum_{X:\mathcal{U}} \sum_{x:X} (\mathsf{fib}_f(b) = X),$$

the first equivalence following from Lemma 4.8.2 (see §4.6 p. 35), the second one from (50), and the third one being straightforward. This proves (49). □

4.9 Exercise 4.4

Statement. (The unstable octahedral axiom.) Suppose $f : A \to B$ and $g : B \to C$ and $b : B$.

(i) Show that there is a natural map $\mathsf{fib}_{g\circ f}(g(b)) \to \mathsf{fib}_g(g(b))$ whose fiber over $(b, \mathsf{refl}_{g(b)})$ is equivalent to $\mathsf{fib}_f(b)$.

(ii) Show that $\mathsf{fib}_{g\circ f}(c) \simeq \sum_{w:\mathsf{fib}_g(c)} \mathsf{fib}_f(\mathsf{pr}_1 w)$.

Solution. (i) Let $b : B$. We define $h : \mathsf{fib}_{g\circ f}(g(b)) \to \mathsf{fib}_g(g(b))$, that is

$$h : \left(\sum_{x:A} \big(g(f(x)) = g(b)\big)\right) \to \sum_{y:B} (g(y) = g(b)),$$

by $h(x,p) :\equiv (f(x), p)$ for $x : A$ and $p : g(f(x)) = g(b)$. We get

$$\begin{aligned}\mathsf{fib}_h(b, \mathsf{refl}_{g(b)}) &:\equiv \sum_{w:\sum_{x:A}(g(f(x))=g(b))} \big(h(w) = (b, \mathsf{refl}_{g(b)})\big)\\ &= \sum_{x:A} \sum_{p:g(f(x))=g(b)} \big((f(x), p) = (b, \mathsf{refl}_{g(b)})\big)\end{aligned}$$

by Exercise 2.10 (see §2.14 p. 24). For each $x : A$ we have

$$\begin{aligned}&\sum_{p:g(f(x))=g(b)} \big((f(x), p) = (b, \mathsf{refl}_{g(b)})\big)\\ &\overset{(a)}{=} \sum_{p:g(f(x))=g(b)} \sum_{q:f(x)=b} \big(\mathsf{transport}^{\lambda(y:B).g(y)=g(b)}(q, p) = \mathsf{refl}_{g(b)}\big)\\ &= \sum_{p:g(f(x))=g(b)} \sum_{q:f(x)=b} \big(\mathsf{ap}_g(q)^{-1} \centerdot p = \mathsf{refl}_{g(b)}\big)\\ &= \sum_{q:f(x)=b} \sum_{p:g(f(x))=g(b)} \big(p = \mathsf{ap}_g(q)\big)\\ &\overset{(b)}{=} \sum_{q:f(x)=b} \mathbf{1}\\ &\overset{(c)}{=} (f(x) = b),\end{aligned}$$

where (a) follows from Theorem 2.7.2 (see §2.1 p. 14), and (b) and (c) follow from §3.4 p. 26.

(ii) We have

$$\begin{aligned}
&\sum_{w:\mathsf{fib}_g(c)} \mathsf{fib}_f(\mathsf{pr}_1 w)\\
\equiv &\sum_{w:\sum_{y:B}(g(y)=c)} \sum_{x:A} \big(f(x) = \mathsf{pr}_1 w\big)\\
\overset{\text{(a)}}{=} &\sum_{y:B} \sum_{q:g(y)=c} \sum_{x:A} \big(f(x) = y\big)\\
= &\sum_{x:A} \sum_{y:B} (g(y) = c) \times (f(x) = y).
\end{aligned}$$

where (a) follows from Exercise 2.10 (see §2.14 p. 24). For each $x : A$ we check that the maps

$$\varphi : \left(\sum_{y:B} (g(y) = c) \times (f(x) = y)\right) \to \big(g(f(x)) = c\big)$$

and

$$\psi : \big(g(f(x)) = c\big) \to \sum_{y:B} (g(y) = c) \times (f(x) = y)$$

defined by

$$\varphi(y, q, p) :\equiv \mathsf{ap}_g(p) \bullet q, \quad \psi(r) :\equiv \big(f(x), r, \mathsf{refl}_{f(x)}\big)$$

are inverses. This gives

$$\left(\sum_{w:\mathsf{fib}_g(c)} \mathsf{fib}_f(\mathsf{pr}_1 w)\right) = \left(\sum_{x:A} \Big(g(f(x)) = c\Big)\right) = \mathsf{fib}_{g\circ f}(c),$$

where the last equality follows from §3.4 p. 26.

4.10 Exercise 4.5

Statement. Prove that equivalences satisfy the 2-out-of-6 property: given $f : A \to B$ and $g : B \to C$ and $h : C \to D$, if $g \circ f$ and $h \circ g$ are equivalences, so are f, g, h, and $h \circ g \circ f$. Use this to give a higher-level proof of Theorem 2.11.1.

Solution. Let $(g \circ f)^{-1}$ be an inverse of $g \circ f$ and $(h \circ g)^{-1}$ an inverse of $h \circ g$. Setting $k :\equiv f \circ (g \circ f)^{-1}$, we get

$$g \circ k = \mathsf{id}_C \tag{51}$$

and

$$\begin{aligned} k &:\equiv f \circ (g \circ f)^{-1} \\ &= \big((h \circ g)^{-1} \circ h \circ g\big) \circ f \circ (g \circ f)^{-1} \\ &= (h \circ g)^{-1} \circ h \circ \big(g \circ f \circ (g \circ f)^{-1}\big) \\ &= (h \circ g)^{-1} \circ h, \end{aligned}$$

which implies

$$k \circ g = (h \circ g)^{-1} \circ h \circ g = \mathsf{id}_B. \tag{52}$$

Now (51) and (52) show that g is invertible. The end of the exercise is straightforward. For the higher-level proof of Theorem 2.11.1, see §2.9. □

4.11 Exercise 4.6 (iii)

Statement. For $A, B : \mathcal{U}$, define

$$\mathsf{idtoqinv}_{A,B} : (A = B) \to \sum_{f:A\to B} \mathsf{qinv}(f)$$

by path induction in the obvious way. Let qinv-univalence denote the modified form of the univalence axiom which asserts that for all $A, B : \mathcal{U}$ the function $\mathsf{idtoqinv}_{A,B}$ has an inverse.

(i) Show that qinv-univalence can be used instead of univalence in the proof of function extensionality in §4.9.

(ii) Show that qinv-univalence can be used instead of univalence in the proof of Theorem 4.1.3.

(iii) Show that qinv-univalence is inconsistent (i.e. allows construction of an inhabitant of $\mathbf{0}$). Thus, the use of a "good" version of isequiv is essential in the statement of univalence.

Solution. Recall that, given a map $f : A \to B$, the type $\mathsf{qinv}(f)$ is defined by

$$\mathsf{qinv}(f) :\equiv \sum_{g:B\to A} (f \circ g) \sim \mathsf{id}_B) \times (g \circ f) \sim \mathsf{id}_A) \tag{53}$$

(see beginning of Chapter 4 in the book), and that $\mathsf{ishae}(f)$ is defined by

$$\mathsf{ishae}(f) :\equiv \sum_{g:B\to A}\ \sum_{\eta:g\circ f\sim\mathsf{id}_A}\ \sum_{\varepsilon:f\circ g\sim\mathsf{id}_B}\ \prod_{x:A} f(\eta x) = \varepsilon(fx) \tag{54}$$

(see beginning of Definition 4.2.1 in the book). (Recall that $\mathsf{ishae}(f)$ stands for "f is a half-adjoint equivalence".)

We leave (i) and (ii) to the reader, and prove (iii). In view of Theorem 4.1.3 it suffices to show that, for any map $f : A \to B$, the type $\mathsf{qinv}(f)$ is a mere proposition.

For any types A and B we put

$$(A \simeq B) :\equiv \sum_{f:A\to B} \mathsf{ishae}(f), \quad (A \backsimeq B) :\equiv \sum_{f:A\to B} \mathsf{qinv}(f).$$

By Theorem 4.2.3 and the comment preceding it, there are dependent functions

$$i : \prod_{A,B:\mathcal{U}}\ \prod_{f:A\to B} \mathsf{ishae}(f) \to \mathsf{qinv}(f), \quad p : \prod_{A,B:\mathcal{U}}\ \prod_{f:A\to B} \mathsf{qinv}(f) \to \mathsf{ishae}(f).$$

For all $A, B : \mathcal{U}$ let $e(A, B) : (A = B) \to (A \backsimeq B)$ be the natural map (which we assume to admit an inverse).

Claim: if $f : A \to B$ and $x : \mathsf{qinv}(f)$, then $i(A, B, f, p(A, B, f, x)) = x$.

Since $(f, x) : A \backsimeq B$, we can assume by qinv-univalence and path induction that

$$B \equiv A, \quad f \equiv \mathsf{id}_A, \quad x \equiv e(A, A, \mathsf{refl}_A),$$

and the claim follows easily from the definition of i, p and e.

For $f : A \to B$ and $x, y : \mathsf{qinv}(f)$ we have $p(A, B, f, x) = p(A, B, f, y)$ because $\mathsf{ishae}(f)$ is a mere proposition, and the claim implies $x = y$, as required. □

5 Chapter 5

5.1 Induction principle for W-types (§5.3)

Let $W :\equiv \underset{a:A}{\mathsf{W}} B(a)$ be a W-type. Recall the following:

We have a constructor

$$\mathsf{sup} : \prod_{a:A} (B(a) \to W) \to W,$$

and the induction principle can be stated as follows:

We assume the recurrence

$$r : \prod_{a:A} (B(a) \to W) \to (B(a) \to E) \to E$$

if we are given a type $E : \mathcal{U}$ (case 1), and

$$r : \prod_{a:A} \prod_{f:B(a)\to W} \left(\prod_{b:B(a)} E(f(b)) \right) \to E(\mathsf{sup}(a, f))$$

if we are given a type family $E : W \to \mathcal{U}$ (case 2).

The solution s to the above recurrence is a function $s : W \to E$ in case 1 and a dependent function $s : \prod_{w:W} E(w)$ in case 2. In both cases the computation rule

$$s(\mathsf{sup}(a, f)) = r(a, f, s \circ f)$$

holds for all $a : A$ and all $f : B(a) \to W$.

5.2 Second bullet of §5.6

This is about the comment containing the second bullet of §5.6 of the book:

$$k : ((D \to \mathsf{Prop}) \to \mathsf{Prop}) \to D.$$

The recursion principle can be stated as follows: Given

$$f : ((D \to \mathsf{Prop}) \to \mathsf{Prop}) \to A,$$

we get $g : D \to A$ such that $g(k(\theta)) \equiv f(\theta)$ for all $\theta : (D \to \mathsf{Prop}) \to \mathsf{Prop}$.

5.3 Before Remark 5.6.3

Just before Remark 5.6.3 we read:

«This is a contradiction: no proposition can be equivalent to its negation. (Supposing $P \Leftrightarrow \neg P$, if P, then $\neg P$, and so $\mathbf{0}$; hence $\neg P$, but then P, and so $\mathbf{0}$.)»

Here are some more details about the above parenthesis:

If $B : A \to \mathcal{U}$ is a type family over a type A, then we have a codiagonal map

$$(\lambda f.\lambda a.f(a,a)) : \left(A \to \prod_{a:A} B(a)\right) \to \prod_{a:A} B(a). \tag{55}$$

In the particular case when $B : \mathcal{U}$ is constant, the above type becomes

$$(A \to A \to B) \to (A \to B).$$

If we assume further $A \equiv P$ and $B \equiv \mathbf{0}$, we get a map $g : (P \to \neg P) \to \neg P$. Recall that we want to prove $(P \Leftrightarrow \neg P) \to \mathbf{0}$. In view of g, our assumption $P \to \neg P$ implies $\neg P$ (and $\neg P$ clearly implies $\mathbf{0}$ under our assumption $P \Leftrightarrow \neg P$). $\square$

5.4 Displays (5.6.4)–(5.6.7)

As Display (5.6.4) of the book, which reads

$$c : (A \to W) \to (B \to C \to W) \to D \to W \to W,$$

can be rewritten as

$$c : (A \to W) \to ((B \times C) \to W) \to D \to (\mathbf{1} \to W) \to W,$$

we can (and do) assume that the constructor c has the form

$$c : (A \to W) \to B \to W.$$

The recurrence takes the form

$$r : (A \to W) \to (A \to P) \to B \to P$$

(see (5.6.5) in the book) if $P : \mathcal{U}$, and

$$r : \prod_{\alpha:A\to W} \left(\prod_{a:A} P(\alpha(a))\right) \to \prod_{b:B} P(c(\alpha, b)).$$

(see (5.6.7) in the book) if $P : W \to \mathcal{U}$. The solution s to the recurrence r satisfies in both cases the computation rule

$$s(c(\alpha, b)) \equiv r(\alpha, s \circ \alpha, b)$$

for all $\alpha : A \to W$ and all $b : B$ (see (5.6.6) in the book).

5.5 Proof of Theorem 5.8.2

Here is a minor rewriting of the proof of the fact that $\mathsf{ppmap}(R,S)$ is a mere proposition in implication (i) $\implies$ (ii) in the proof of Theorem 5.8.2:

Let

$$f,g:\prod_{b:A}R(b)\to S(b),\qquad p:f(a_0,r_0)=s_0,\qquad q:g(a_0,r_0)=s_0,$$

so that

$$(f,p),(g,q):\mathsf{ppmap}(R,S).$$

It suffices to prove $(f,p)=(g,q)$. Setting

$$D(b,r):\equiv\big(f(b,r)=g(b,r)\big),\quad d:\equiv p\bullet q^{-1},$$

we get $h:f\sim g$ with $h(a_0,r_0)=p\bullet q^{-1}$. Function extensionality yields a path $t:f=g$ satisfying

$$t_\star(p)=h(a_0,r_0)^{-1}\bullet p=(p\bullet q^{-1})^{-1}\bullet p=q.\ \square$$

5.6 Proof of Theorem 5.8.4 (iii)

By Theorem 2.15.7 (see §4.2 p. 33) the type

$$\sum_{g:\prod_{(a,b:A)}R(a,b)\to S(a,b)}\ \prod_{a:A}\Big(g\big(a,a,r_0(a)\big)=s_0(a)\Big)$$

is equivalent to the type

$$\prod_{a:A}\ \sum_{g(a):\prod_{(b:a)}R(a,b)\to S(a,b)}\Big(g\big(a,a,r_0(a)\big)=s_0(a)\Big).\ \square$$

6 Chapter 6

6.1 Display (6.2.2)

If P is a type family over a type A, if $f:\prod_{a:A}P(a)$ is a dependent function, and if $p:x=y$ is a path in A, then we have

$$\mathsf{apd}_f(p):f(x)=^P_p f(y),$$

i.e. $\mathsf{apd}_f(p)$ is a dependent path from $f(x)$ to $f(y)$ over p.

6.2 Lemma 6.4.1

Recall the statement of the lemma: *The type* $\mathbb{S}^1$ *satisfies* $\mathsf{loop} \neq \mathsf{refl}_{\mathsf{base}}$.

This implies that $\mathbb{S}^1$ is not a set, and thus, by Lemma 3.3.4 (which says that every mere proposition is a set), that $\mathbb{S}^1$ is not a mere proposition, and thus, by §3.4 p. 26, that $\mathbb{S}^1$ is not contractible.

6.3 Proof of Corollary 6.4.3

For the reader's convenience we paste the statement and the proof of Corollary 6.4.3:

Corollary 6.4.3. *If the type* $\mathbb{S}^1$ *belongs to some universe* $\mathcal{U}$*, then* $\mathcal{U}$ *is not a 1-type.*

Proof. The type $\mathbb{S}^1 = \mathbb{S}^1$ in $\mathcal{U}$ is, by univalence, equivalent to the type $\mathbb{S}^1 \simeq \mathbb{S}^1$ of autoequivalences of $\mathbb{S}^1$, so it suffices to show that $\mathbb{S}^1 \simeq \mathbb{S}^1$ is not a set. For this, it suffices to show that its equality type $\mathsf{id}_{\mathbb{S}^1} =_{(\mathbb{S}^1 = \mathbb{S}^1)} \mathsf{id}_{\mathbb{S}^1}$ is not a mere proposition. Since being an equivalence is a mere proposition, this type is equivalent to $\mathsf{id}_{\mathbb{S}^1} =_{(\mathbb{S}^1 \to \mathbb{S}^1)} \mathsf{id}_{\mathbb{S}^1}$. But by function extensionality, this is equivalent to $\prod_{x:\mathbb{S}^1}(x = x)$, which as we have seen in Lemma 6.4.2 contains two unequal elements. □

The proof of Corollary 6.4.3 uses the following claim:

Claim. *Let A and B be types; let*

$$\varphi_1 : (A \simeq B) \to (A \to B), \quad \varphi_2 : \prod_{u:A\simeq B} \mathsf{ishae}(\varphi_1(u))$$

be the first and second projection (see (54) *p. 42 for the definition of* ishae*); and let $u, v : A \simeq B$. Then the map*

$$\mathsf{ap}_{\varphi_1} : (u = v) \to (\varphi_1(u) = \varphi_1(v)) \tag{56}$$

is invertible.

Proof. By Theorem 4.4.5 it suffices to show that the fibers of (56) are contractible. Set

$$C :\equiv \sum_{p:\varphi_1(u)=\varphi(v)} \Big(p_\star(\varphi_2(u)) =_{\mathsf{ishae}(f)} \varphi_2(v)\Big)$$

and let $\psi_1 : C \to (\varphi_1(u) = \varphi(v))$ be the first projection. By Theorem 2.7.2 (see §2.1 p. 14) there is an invertible map $f : (u = v) \to C$ such that $\psi_1 \circ f = \mathsf{ap}_{\varphi_1}$.

Let $p : \varphi_1(u) = \varphi_1(v)$. We have $\mathsf{fib}_{\mathsf{ap}_{\varphi_1}}(p) \simeq \mathsf{fib}_{\psi_1}(p)$. By Lemma 4.8.1 (see §4.5 p. 34) the latter is equivalent to

$$p_\star(\varphi_2(u)) =_{\mathsf{ishae}(f)} \varphi_2(v). \tag{57}$$

As $\mathsf{ishae}(f)$ is contractible by Theorem 4.2.13, the results of §3.4 p. 26 imply that it is a mere proposition, and Lemma 3.3.4 (which says that every mere proposition is a set) implies that it is a set. In particular (57) is contractible. □

6.4 Proof of Lemma 6.5.1

The sentence containing the first display follows from Theorem 2.11.3.

6.5 Section 6.8

Here is the *recursion principle* for the pushout

$$\boxed{Z :\equiv A \sqcup^C B}$$

attached to the diagram

$$\begin{tikzcd}
C \arrow[r, "g"] \arrow[d, "f"'] & B \arrow[d, "\mathsf{inr}"] \\
A \arrow[r, "\mathsf{inl}"'] \arrow[ur, Rightarrow, "\mathsf{glue}"] & Z,
\end{tikzcd}$$

where glue is a homotopy from $\mathsf{inl} \circ f$ to $\mathsf{inr} \circ g$:

Given

$$\alpha : A \to D, \quad \beta : B \to D, \quad \gamma : \alpha \circ f \sim \beta \circ g$$

there is an $s : Z \to D$ such that

$$s \circ \mathsf{inl} \equiv \alpha, \quad s \circ \mathsf{inr} \equiv \beta, \quad \mathsf{ap}_s(\mathsf{glue}(c)) = \gamma(c)$$

for all $c : C$.

Here is the *induction principle*: Let $D : Z \to \mathcal{U}$ be a type family, let

$$\alpha : \prod_{a:A} D(\mathsf{inl}(a)), \quad \beta : \prod_{b:B} D(\mathsf{inr}(b)),$$

$$\gamma : \prod_{c:C} \Big(\alpha\big(f(c)\big) =^{D}_{\mathsf{glue}(c)} \beta\big(g(c)\big)\Big).$$

Then there is an $s : \prod_{z:Z} D(z)$ such that

$$s \circ \mathsf{inl} \equiv \alpha, \quad s \circ \mathsf{inr} \equiv \beta, \quad \mathsf{apd}_s(\mathsf{glue}(c)) = \gamma(c)$$

for all $c : C$.

Here is the *uniqueness principle*: Let $h, h' : Z \to D$ and

$$\alpha : \prod_{a:A} \Big(h\big(\mathsf{inl}(a)\big) = h'\big(\mathsf{inl}(a)\big)\Big), \quad \beta : \prod_{b:B} \Big(h\big(\mathsf{inr}(b)\big) = h'\big(\mathsf{inr}(b)\big)\Big),$$

$$\gamma : \prod_{c:C} \Big(\alpha\big(f(c)\big) =^{z \mapsto (h(z)=h'(z))}_{\mathsf{glue}(c)} \beta\big(g(c)\big)\Big).$$

Then there is an $s : \prod_{z:Z} \big(h(z) = h'(z)\big)$ such that

$$s \circ \mathsf{inl} \equiv \alpha, \quad s \circ \mathsf{inr} \equiv \beta, \quad \mathsf{apd}_s(\mathsf{glue}(c)) = \gamma(c)$$

for all $c : C$.

⋆

In the proof of Lemma 6.8.2 let us display the definition of $t \circ c_{\sqcup} : \mathsf{cocone}_{\mathcal{D}}(E)$ and of $\mathsf{s}(c) : Z \to E$:

$$t \circ c_{\sqcup} :\equiv (t \circ \mathsf{inl}, t \circ \mathsf{inr}, \mathsf{ap}_t \circ \mathsf{glue}),$$

$$\mathsf{s}(c) \circ \mathsf{inl} :\equiv i, \quad \mathsf{s}(c) \circ \mathsf{inr} :\equiv j, \quad \mathsf{ap}_{\mathsf{s}(c)} \circ \mathsf{glue} := h.$$

(Recall: $t : Z \to E,\ c = (i, j, h) : \mathsf{cocone}_{\mathcal{D}}(E)$.)

Some more details about the end of the proof of Lemma 6.8.2:

Set $u :\equiv \mathsf{s}(t \circ c_{\sqcup})$, and let $z : Z,\ x : C$. We want to show $u = t$ using the uniqueness principle. This easily reduces to verifying that

$$\mathsf{refl}_{t(\mathsf{inl}(f(x)))} =^{z \mapsto (u(z)=t(z))}_{\mathsf{glue}(x)} \mathsf{refl}_{t(\mathsf{inr}(g(x)))},$$

or, equivalently,

$$\mathsf{transport}^{z \mapsto (u(z)=t(z))}(\mathsf{glue}(x), \mathsf{refl}_{t(\mathsf{inl}(f(x)))}) = \mathsf{refl}_{t(\mathsf{inr}(g(x)))},$$

is inhabited. We have, for $z_1, z_2 : Z,\ p : z_1 = z_2$ and $q : u(z_1) = t(z_1)$,

$$\mathsf{transport}^{z \mapsto (u(z)=t(z))}(p, q) = \mathsf{ap}_u(p)^{-1} \cdot q \cdot \mathsf{ap}_t(p).$$

(The above formula is easily proved by induction on p.) Hence it suffices to show $\mathsf{ap}_u(\mathsf{glue}(x)) = \mathsf{ap}_t(\mathsf{glue}(x))$. But this follows immediately from the definition of u. □

6.6 Section 6.9

The induction principle for $\|A\|$ is stated as follows:

"Given any $B : \|A\| \to \mathcal{U}$ together with

- a function $g : \prod_{a:A} B(|a|)$, and
- for any $x, y : \|A\|$ and $u : B(x)$ and $v : B(y)$, a dependent path $q : u =^B_{p(x,y)} v$, where $p(x,y)$ is the path coming from the second constructor of $\|A\|$,

there exists

$$f : \prod_{x:\|A\|} B(x)$$

such that $f(|a|) \equiv g(a)$ for $a : A$, and also another computation rule. However, because there can be at most one function between any two mere propositions (up to homotopy), this induction principle is not really useful (see also Exercise 3.17[1])."

Let us spell out the last sentence above.

We can rewrite the second premise as

($\star$) for any $x, y : \|A\|$, any $u : B(x)$ and any $v : B(y)$, we are given a dependent path $q(x,y,u,v) : u =^B_{p(x,y)} v$, where $p(x,y) : x = y$ is the path coming from the second constructor of $\|A\|$.

Consider the condition

($\star\star$) for any $x : \|A\|$ the type $B(x)$ is a mere proposition.

We claim

$$(\star) \iff (\star\star). \tag{58}$$

Proof of (58). Indeed, assume ($\star$). Letting $x : \|A\|$ and $u, v : B(x)$, we have $r : \mathsf{refl}_x = p(x,x)$ (because $\|A\|$ is a mere proposition and thus a set) and

$$u = (\mathsf{refl}_x)_\star(u) = p(x,x)_\star(u) = v,$$

the second and third paths being respectively $\mathsf{ap}_{s \mapsto s_\star(u)}(r)$ and $q(x,x,u,v)$. This proves ($\star\star$). The converse is clear. □

We have in particular the following induction principle:

Given any $P : \|A\| \to \mathsf{Prop}$ together with a function $f : \prod_{a:A} P(|a|)$, there exists $g : \prod_{x:\|A\|} P(x)$ such that $g(|a|) \equiv f(a)$ for $a : A$.

[1] See §3.11 p. 30.

6.7 Notion of quotient (beginning of §6.10)

We shall define quotients via truncations in §7.5 below.

6.8 Proof of Theorem 6.10.6

Recall the statement of Theorem 6.10.6:

For any equivalence relation R on A, the type $A/\!\!/R$ is equivalent to the set-quotient A/R.

The proof uses implicitly the fact that $A/\!\!/R$ is a set. This fact follows immediately from Theorems 7.1.8, 7.1.9 and 7.1.11 (see below) of the book. The reader can check that the proof of Theorem 6.10.6 doesn't use Theorems 7.1.8, 7.1.9 and 7.1.11.

For the reader's convenience we paste the statement of Theorems 7.1.8, 7.1.9 and 7.1.11:

Theorem 7.1.8. *Let $n \geq -2$, and let $A : \mathcal{U}$ and $B : A \to \mathcal{U}$. If A is an n-type and for all $a : A$, $B(a)$ is an n-type, then so is $\sum_{x:A} B(x)$.*

Theorem 7.1.9. *Let $n \geq -2$, and let $A : \mathcal{U}$ and $B : A \to \mathcal{U}$. If for all $a : A$, $B(a)$ is an n-type, then so is $\prod_{x:A} B(x)$.*

Theorem 7.1.11. *For any $n \geq -2$, the type n-Type is an $(n+1)$-type.*

6.9 Lemma 6.10.8

Here is a corollary to Lemma 6.10.8:

In the setting of Lemma 6.10.8, let us write q' for the map from A to $(A/\!\sim)$ denoted by q in Lemma 6.10.8. Then there are inverse maps f and g such that the following diagram commutes:

$$\begin{tikzcd}
 & A \arrow[ld, "q"'] \arrow[rd, "q'"] & \\
A/\!\sim \arrow[rr, shift left, "f"] & & (A/\!\sim) \arrow[ll, shift left, "g"]
\end{tikzcd}$$

6.10 After Lemma 6.12.1

Let us spell out the induction principle for the higher inductive type W defined right after Lemma 6.12.1:

Recall that we are given $f, g : B \to A$, and that the constructors defining W are

$$\mathsf{c} : A \to W, \quad \mathsf{p} : \prod_{b:B} \Big(\mathsf{c}(f(b)) = \mathsf{c}(g(b))\Big).$$

Given a type family $P : W \to \mathcal{U}$, the solution $s : \prod_{w:W} P(w)$ to the recurrences

$$h : \prod_{a:A} P(\mathsf{c}(a)), \quad q : \prod_{b:B} \Big(h(f(b)) =^{P}_{\mathsf{p}(b)} h(g(b))\Big)$$

satisfies the computation rules

$$s(\mathsf{c}(a)) :\equiv h(a), \quad \mathsf{apd}_s(\mathsf{p}(b)) := q(b)$$

for all $a : A$ and all $b : B$.

We now spell out the induction principle for the higher inductive type $\widetilde{W}$ defined right before Lemma 6.12.2 (*the Flattening Lemma*):

First recall the following:

We suppose, in addition to the above items,

- $C : A \to \mathcal{U}$ is a family of types over A, and
- $D : \prod_{b:B} C(f(b)) \simeq C(g(b))$ is a family of equivalences over B,

we define a type family $P : W \to \mathcal{U}$ recursively by

$$P(\mathsf{c}(a)) :\equiv C(a), \quad P(\mathsf{p}(b)) := \mathsf{ua}(D(b)),$$

and we let $\widetilde{W}$ be the higher inductive type generated by

- $\widetilde{\mathsf{c}} : \prod_{a:A} C(a) \to \widetilde{W}$ and
- $\widetilde{\mathsf{p}} : \prod_{b:B} \prod_{y:C(f(b))} \Big(\widetilde{\mathsf{c}}(f(b), y) =_{\widetilde{W}} \widetilde{\mathsf{c}}(g(b), D(b)(y))\Big)$.

Then the induction principle for the higher inductive type $\widetilde{W}$ takes the following form:

If

$$Q : \left(\sum_{x:W} P(x) \right) \to \mathcal{U}$$

is a type family, then the solution

$$k : \prod_{z:\sum_{w:W} P(w)} Q(z)$$

to the recurrences

$$c : \prod_{a:A} \prod_{x:C(a)} Q(\widetilde{\mathsf{c}}(a,x))$$

and

$$p : \prod_{b:B} \prod_{y:C(f(b))} \left(\widetilde{\mathsf{p}}(b,y)_\star \, c\Big(f(b), y\Big) = c\Big(g(b), D(b)(y)\Big) \right)$$

satisfies

$$k(\widetilde{\mathsf{c}}(a,x)) \equiv c(a,x)$$

for all $a : A$ and all $x : C(a)$, as well as

$$\mathsf{apd}_k(\widetilde{\mathsf{p}}(b,y)) = p(b,y)$$

for all $b : B$ and all $y : C(f(b))$.

6.11 Exercise 6.9

Recall the statement:

Assuming the Law of Excluded Middle, construct a dependent function $f : \prod_{A:\mathcal{U}} A \to A$ such that $f(\mathbf{2}) : \mathbf{2} \to \mathbf{2}$ is the nonidentity automorphism.

Recall that the Law of Excluded Middle is defined in (3.4.1) by

$$\mathsf{LEM} : \prod_{A:\mathcal{U}} \mathsf{isProp}(A) \to (A + \neg A).$$

I learned the following argument from Jason Gross; see

https://groups.google.com/forum/#!topic/hott-cafe/Pp7AgvKr5PI.

Let A be a type. We must define $f(A) : A \to A$. We set

$$C :\equiv \sum_{f:A \to A} \mathsf{isequiv}(f) \times \neg(f = \mathsf{id}_A).$$

In particular we have the first projection $\mathsf{pr}_1 : C \to A \to A$. By §3.4 p. 26 we also have

$$g : \prod_{B:\mathcal{U}} \mathsf{isProp}(\mathsf{isContr}(B)).$$

We define

$$h_0 : \neg\mathsf{isContr}(C) \to A \to A, \qquad h_1 : \mathsf{isContr}(C) \to A \to A$$

by setting $h_0(x) :\equiv \mathsf{id}_A$ for all $x : \neg\mathsf{isContr}(C)$, and by letting $h_1(x)$ be the first projection of the center of x, for all $x : \mathsf{isContr}(C)$. This induces a map

$$h : \big(\mathsf{isContr}(C) + \neg\mathsf{isContr}(C)\big) \to A \to A.$$

We finally set

$$f(A) :\equiv h\Big(\mathsf{LEM}\big(\mathsf{isContr}(C), g(C)\big)\Big).$$

It is straightforward to check that this definition does the job. □

7 Chapter 7

7.1 Theorem 7.1.8

Here is a corollary to Theorem 7.1.8:

The fibers of a map between n-types are n-types.

For the reader's convenience we paste the statement of Theorem 7.1.8:

Theorem 7.1.8. *Let $n \geq -2$, and let $A : \mathcal{U}$ and $B : A \to \mathcal{U}$. If A is an n-type and for all $a : A$, $B(a)$ is an n-type, then so is $\sum_{x:A} B(x)$.*

7.2 First proof of Theorem 7.2.2

We must prove

Claim. *If $R : X \to X \to \mathsf{Prop}$ is a mere relation on a type X and we have*

$$\rho : \prod_{x:X} R(x,x), \quad f : \prod_{x,y:X} R(x,y) \to (x = y),$$

then X is a set.

Proof. We prove

$$f(x,x,\rho(x)) \cdot p = f(x,y,\mathsf{transport}^{R(x)}(p,\rho(x))) \tag{59}$$

for $p : x = y$ by path induction. For $q : x = y$ we have

$$\begin{aligned} q &= f(x,x,\rho(x))^{-1} \cdot f(x,y,\mathsf{transport}^{R(x)}(q,\rho(x))) \\ &= f(x,x,\rho(x))^{-1} \cdot f(x,y,\mathsf{transport}^{R(x)}(p,\rho(x))) \\ &= p, \end{aligned}$$

where the second equality follows from the fact that $R(x,y)$ is a mere proposition, and the other equalities follow from (59). □

7.3 Lemma 7.2.8

Recall the statement of Lemma 7.2.8;

Given $n \geq -1$ and $X : \mathcal{U}$. If, given any inhabitant of X it follows that X is an n-type, then X is an n-type.

Proof. Set $A :\equiv X$ and

$$B(a) :\equiv \prod_{x:X} \mathsf{is}\text{-}(n-1)\text{-}\mathsf{type}(a = x)$$

in the definition of the codiagonal map (55) p. 44. □

7.4 Induction principle for n-truncations (§7.3)

Recall that A is an arbitrary type and that $n : \mathbb{N}$ satisfies $n \geq -1$.

We spell out the induction principle for the higher inductive type $\|A\|_n$.

Let $\|A\|_n$ be defined by the three constructors

$$c_1 : A \to \|A\|_n, \quad c_2 : (B \to \|A\|_n) \to \|A\|_n, \quad c_3 : \prod_{d:B\to\|A\|_n} \prod_{b:B} \big(d(b) = c_2(d)\big).$$

Let $P : \|A\|_n \to \mathcal{U}$ be a type family. We say that a dependent function $f : \prod_{w:\|A\|_n} P(w)$ solves the three recurrences

$$r_1 : \prod_{a:A} P(c_1(a)),$$

$$r_2 : \prod_{d:B\to\|A\|_n} \left(\prod_{b:B} P(d(b))\right) \to P(c_2(d)),$$

$$r_3 : \prod_{d:B\to\|A\|_n} \prod_{e:\prod_{b:B} P(d(b))} \prod_{b:B} \big(e(b) =^P_{c_3(d,b)} r_2(d,e)\big)$$

if the computation rules

$$f(c_1(a)) \equiv r_1(a), \quad f(c_2(d)) \equiv r_2(d, f\circ d), \quad \mathsf{apd}_f(c_3(d,b)) = r_3(d, f\circ d, b)$$

hold for all $a : A$, all $d : B \to \|A\|_n$ and all $b : B$.

(In the book the last two computational rules are omitted.)

7.5 Defining quotients via truncations

Let $R : A \to A \to \mathsf{Prop}$ be a mere relation on a type A. Let $A/'R$ be the higher inductive type whose constructors are

$$q' : A \to A/'R, \quad c : \prod_{a,b:A} R(a,b) \to (q'(a) = q'(b)).$$

In particular, the induction principle reads as follows: Given a type family $P : A/'R \to \mathcal{U}$, we say that a dependent function $f : \prod_{x:A/'R} P(x)$ satisfies the recurrences

$$r : \prod_{a:A} P(q'(a)), \quad s : \prod_{a,b:A} \prod_{u:R(a,b)} \Big(r(a) =^P_{c(a,b,u)} r(b)\Big)$$

if the computation rules

$$f(q'(a)) \equiv r(a), \quad \mathsf{apd}_f(c(a,b,u)) = s(a,b,u)$$

hold for all $a, b : A$ and all $u : R(a,b)$.

Now we can define the quotient A/R by $A/R :\equiv \|A/'R\|_0$.

7.6 Proof of Theorem 7.3.2

Claim. *Let $n \geq -1$ and let A be a type. Then A is an n-type if and only if the type*

$$\mathsf{Map}_\star\big((\mathbb{S}^{n+1}, \mathsf{base}), (A, a)\big)$$

is contractible for all $a : A$.

Proof. Use Theorem 7.2.9 (see below) and the equivalence

$$\mathsf{Map}_\star\big((\mathbb{S}^{n+1}, \mathsf{base}), (A, a)\big) \simeq \Omega^{n+1}(A, a)$$

at the end of §6.5 of the book. □

We verify the existence of u and v in the proof of Theorem 7.3.2 as follows. We define $t : \mathbb{S}^{n+1} \to P(h(r))$ as in the book, we set $h'(r, r') :\equiv u :\equiv t(\mathsf{base})$, we define $c_u : \mathbb{S}^{n+1} \to P(h(r))$ by $c_u(x) :\equiv u$ for all $x : \mathbb{S}^{n+1}$, and we set

$$M :\equiv \mathsf{Map}_\star\Big(\big(\mathbb{S}^{n+1}, \mathsf{base}\big), \big(P(h(r)), u\big)\Big).$$

Since $(t, \mathsf{refl}_u), (c_u, \mathsf{refl}_u) : M$ and M is contractible, there is a path $v : t = c_u$. □

For the reader's convenience we paste the statement of Theorem 7.2.9:

Theorem 7.2.9. *For every $n \geq -1$, a type A is an n-type if and only if $\Omega^{n+1}(A, a)$ is contractible for all $a : A$.*

7.7 Proof of Theorem 7.3.5

Theorem 7.3.5 (see below) can also be proved by path induction.

For the reader's convenience we paste the statement of Theorem 7.3.5:

Theorem 7.3.5. *Given $f, g : A \to B$ and a homotopy $h : f \sim g$, there is an induced homotopy $\|h\|_n : \|f\|_n \sim \|g\|_n$ such that the composite*

$$|f(a)|_n \xlongequal{\mathsf{nat}^f_n(a)^{-1}} \|f\|_n(|a|_n) \xlongequal{\|h\|_n(|a|_n)} \|g\|_n(|a|_n) \xlongequal{\mathsf{nat}^g_n(a)} |g(a)|_n \tag{60}$$

is equal to $\mathsf{ap}_{|-|_n}(h(a))$.

7.8 Proof of Theorem 7.3.12

The proof uses various times the principle of double induction for n-truncation, principle which we spell out here.

Recall the simple induction principle:

To each couple (P, f) with $P : \|A\|_n \to n\text{-}\mathsf{Type}$ and $f : \prod_{x:A} P(|x|_n)$ is attach a section $f' : \prod_{u:\|A\|_n} P(u)$ satisfying $f'(|x|_n) \equiv f(x)$ for all $x : A$.

The double induction principle says:

To each couple (P, f) with

$$P : \|A\|_n \to \|A\|_n \to n\text{-}\mathsf{Type}$$

and $f : \prod_{x,y:A} P(|x|_n, |y|_n)$ is attach a section $f'' : \prod_{u,v:\|A\|_n} P(u, v)$ satisfying $f''(|x|_n, |y|_n) \equiv f(x, y)$ for all $x, y : A$.

We derive the double induction principle from the simple induction principle as follows. From

$$f : \prod_{x:A} \left(\prod_{y:A} P(|x|_n, |y|_n) \right)$$

we get

$$f' : \prod_{u:\|A\|_n} \left(\prod_{y:A} P(u, |y|_n) \right)$$

satisfying $f'(|x|_n)(y) \equiv f(x, y)$ for all $x, y : A$. From $f'(u) : \prod_{y:A} P(u, |y|_n)$ we get $f'(u)' : \prod_{v:\|A\|_n} P(u, v)$ satisfying $f'(u)'(|y|_n) \equiv f'(u, y)$ for all $u : \|A\|_n$ and all $y : A$. As we have

$$f'(|x|_n)'(|y|_n) \equiv f'(|x|_n)(y) \equiv f(x, y)$$

for all $x, y : A$, we can set $f''(u, v) :\equiv f'(u)'(v)$ for all $u, v : \|A\|_n$. $\square$

It seems to me that the equality

$$\mathsf{encode}(\mathsf{decode}(|p|_n)) = |p|_n$$

can be proved by path induction on $p : x = y$.

7.9 Paths between cocones

The following observation might be made just after Definition 7.4.2:

If (i, j, h) and (i', j', h') are cocones under $\mathcal{D}$ with base D, then a path $(i, j, h) = (i', j', h')$ is given by attaching to each $c : C$ a path $p(c) : i(f(c)) = i'(f(c))$, a path $q(c) : j(g(c)) = j'(g(c))$, and a path $r(c) : p(c) \cdot h'(c) = h(c) \cdot q(c)$:

$$\begin{tikzcd}
i(f(c)) \arrow[r, equal, "h(c)"] \arrow[d, equal, "p(c)"'] & j(g(c)) \arrow[d, equal, "q(c)"] \\
i'(f(c)) \arrow[r, equal, "h'(c)"'] & j'(g(c)).
\end{tikzcd}$$

7.10 Definition 7.4.7

The setting can be summarized by the diagram

$$\begin{tikzcd}
C \arrow[r, "g"] \arrow[d, "f"'] & B \arrow[d, "j"] \\
A \arrow[r, "i"'] \arrow[ur, Rightarrow, "h"] & D,
\end{tikzcd}$$

where the double arrow represents a homotopy $h : i \circ f \sim j \circ g$.

Definition 7.4.7 contains the judgment

$$\|h\|_n : \|i\|_n \circ \|f\|_n \sim \|j\|_n \circ \|g\|_n.$$

(See (60) p. 56.) In fact we have $\|h\|_n : \|i \circ f\|_n \sim \|j \circ g\|_n$. Here is a fix.

Given the commutative diagram

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[d] & B \arrow[r, "g"] \arrow[d] & C \arrow[d] \\
\|A\|_n \arrow[r, "\|f\|_n"'] & \|B\|_n \arrow[r, "\|g\|_n"'] & \|C\|_n,
\end{tikzcd}$$

where the vertical maps are the obvious ones, we define

$$\varphi_n^{g,f} : \|g\|_n \circ \|f\|_n \sim \|g \circ f\|_n$$

by the commutative square

$$\begin{tikzcd}
\|g\|_n \|f\|_n |a|_n \arrow[r, equal, "\mathsf{ap}_{\|g\|_n}(\mathsf{nat}_n^f a)"] \arrow[d, equal, "\varphi_n^{g,f}(|a|_n)"'] & \|g\|_n |fa|_n \arrow[d, equal, "\mathsf{nat}_n^g(fa)"] \\
\|gf\|_n |a|_n \arrow[r, equal, "\mathsf{nat}_n^{gf}(a)"'] & |gfa|_n,
\end{tikzcd}$$

that is, we set

$$\varphi_n^{g,f}(|a|_n) :\equiv \mathsf{ap}_{\|g\|_n}(\mathsf{nat}_n^f a) \bullet \mathsf{nat}_n^g(fa) \bullet (\mathsf{nat}_n^{gf}(a))^{-1}.$$

(We assume $a : A$, and omit most of the parenthesis.)

Going back to Definition 7.4.7, we may define $\|c\|_n$ as being $(\|i\|_n, \|j\|_n, h'_n)$, where h'_n is the homotopy obtained by composing the three homotopies

$$\|i\|_n \circ \|f\|_n \overset{\varphi_n^{i,f}}{\sim} \|i \circ f\|_n \overset{\|h\|_n}{\sim} \|j \circ g\|_n \overset{(\varphi_n^{j,g})^{-1}}{\sim} \|j\|_n \circ \|g\|_n.$$

7.11 Equality (7.4.11)

The setting can be summarized by the diagram

$$\begin{tikzcd}
C \arrow[r, "g"] \arrow[d, "f"'] & B \arrow[d, "j"] \\
A \arrow[r, "i"'] \arrow[ur, Rightarrow, "h"] & D,
\end{tikzcd}$$

where the double arrow represents a homotopy $h : i \circ f \sim j \circ g$. Equality (7.4.11) reads $|-|_n^D \circ c = \|c\|_n \circ |-|_n^{\mathcal{D}}$, and, by §7.9 p. 57, is equivalent to

$$\Big(|-|_n^D \circ i\ ,\ |-|_n^D \circ j\ ,\ \mathsf{ap}_{|-|_n^D} \circ h\Big) = \Big(\|i\|_n \circ |-|_n^A\ ,\ \|j\|_n \circ |-|_n^B\ ,\ k\Big), \tag{61}$$

where k is the homotopy from $\|i\|_n \circ |-|_n^A \circ f$ to $\|j\|_n \circ |-|_n^B \circ g$ such that $k(z)$ is, for all $z : C$, the composite path

$$\|i\||fz| \overset{\mathsf{ap}_{\|i\|}(\mathsf{nat}^f z)^{-1}}{=\!=\!=\!=\!=} \|i\|\|f\||z| \overset{h'|z|}{=\!=\!=\!=\!=} \|j\|\|g\||z| \overset{\mathsf{ap}_{\|j\|}(\mathsf{nat}^g z)}{=\!=\!=\!=\!=} \|j\||gz|.$$

(Here and in the sequel of this section we drop the subscripts n, the superscripts A, B, C, D, and most of the parenthesis.) In turn, (61) is equivalent to the commutativity of the diagram

$$\begin{tikzcd}
\|i\||fz| \arrow[rrr, equal, "kz"] \arrow[d, equal, "\mathsf{refl}"'] & & & \|j\||gz| \arrow[d, equal, "\mathsf{refl}"] \\
\|i\||fz| \arrow[r, equal, "\mathsf{ap}_{\|i\|}(\mathsf{nat}^f z)^{-1}"] \arrow[d, equal, "\mathsf{nat}^i(fz)"'] & \|i\|\|f\||z| \arrow[r, equal, "h'|z|"] \arrow[d, equal, "\varphi^{\|i\|,\|f\|}(|z|)"'] & \|j\|\|g\||z| \arrow[r, equal, "\mathsf{ap}_{\|j\|}(\mathsf{nat}^g z)"] \arrow[d, equal, "\varphi^{\|j\|,\|g\|}(|z|)"] & \|j\||gz| \arrow[d, equal, "\mathsf{nat}^j(gz)"] \\
|ifz| \arrow[r, equal, "\mathsf{nat}^{if}(z)^{-1}"'] \arrow[d, equal, "\mathsf{refl}"'] & \|if\||z| \arrow[r, equal, "\|h\||z|"'] & \|jg\||z| \arrow[r, equal, "\mathsf{nat}^{jg}(z)"'] & |jgz| \arrow[d, equal, "\mathsf{refl}"] \\
|ifz| \arrow[rrr, equal, "\mathsf{ap}_{|-|}(hz)"'] & & & |jgz|
\end{tikzcd}$$

for all $z : C$. The upper rectangle commutes by definition of k, the small squares commute by §7.10 above, and the lower rectangle commutes by Lemma 7.3.5 in the book (see (60) p. 56).

7.12 Definition 7.5.1

The following observation, which is an immediate consequence of Corollary 7.3.7, can be made right after Definition 7.5.1:

Let A be a type and let $n : \mathbb{N}$. Then A is contractible if and only if A is an n-connected n-type.

7.13 Proof of Lemma 7.5.10

Here is the statement of Lemma 7.5.10:

Lemma 7.5.10. *Let B be an n-type and let $f : A \to B$ be a function. Then the induced function $g : \|A\|_n \to B$ is an equivalence if and only if f is n-connected.*

Here is a pasting of the proof given in the book, proof to which the two parenthesis has been added:

Proof. By Corollary 7.5.8, $|-|_n$ is n-connected. Thus, since $f = g \circ |-|_n$, by Lemma 7.5.6 f is n-connected if and only if g is n-connected. But since g is a function between n-types, its fibers are also n-types (see §7.1 p. 53 above). Thus, g is n-connected if and only if it is invertible (see §7.12 above).

7.14 Proof of Lemma 7.5.11

The sentence

"Then P is a family of $(n-1)$-types and we have $P(a_0)$; hence we have $\prod_{(a:A)} P(a)$ since $a_0 : \mathbf{1} \to A$ is $(n-1)$-connected"

follows from Lemma 7.5.7, implication (i) ⇒ (iii), in the book.

7.15 Comment before Lemma 7.5.12

The comment

In particular, a pointed type (A, a_0) is 0*-connected if and only if $a_0 : \mathbf{1} \to A$ is surjective, which is to say $\prod_{x:A} \|x = a_0\|$*

before Lemma 7.5.12 implies that the pointed circle $(\mathbb{S}^1, \mathsf{base})$ is 0-connected.

7.16 Lemma 7.6.5

In the last display of the statement of Lemma 7.6.5, the identification $\overline{E}(H, a)$ is a path in the fiber $\mathsf{fib}_{h_2}(h_1(g_1(a)))$ of h_2 over $h_1(g_1(a))$.

8 Chapter 8

8.1 Proof of Theorem 8.2.1

The equality $\| \|A\|_{n+1} \|_n = \|A\|_n$ follows from Lemma 7.3.15.

8.2 Proof of Lemma 8.3.2 and Corollary 8.4.8

The equality $\|\Omega^k(A,a)\|_0 = \Omega^k(\|(A,a)\|_k)$ in the proof of Lemma 8.3.2 follows from Corollary 7.3.14. A similar argument is used in the proof of Corollary 8.4.8. The proof of Corollary 8.4.8 also uses Lemma 7.3.15.

8.3 Lemma 8.5.9

Recall the statement:

Lemma 8.5.9. *The operation of join is associative: if A, B and C are three types then we have an equivalence $(A * B) * C \simeq A * (B * C)$.*

Claim. *For all $a{:}A$, $b{:}B$, $c{:}C$ there is a path*

$$\mathsf{glue}(a, \mathsf{inl}(b))^{-1} \cdot \mathsf{glue}(a, \mathsf{inr}(c)) = \mathsf{inr}(\mathsf{glue}(b,c))$$

in $\mathsf{inr}(\mathsf{inl}(b)) =_{A*(B*C)} \mathsf{inr}(\mathsf{inr}(c))$.

Proof of the claim. Define $P : B * C \to \mathcal{U}$ by $P(x) :\equiv (\mathsf{inl}(a) =_{A*(B*C)} \mathsf{inr}(x))$. We get successively

$$\mathsf{glue}(a, -) : \prod_{x:B*C} P(x),$$

$$\mathsf{apd}_{\mathsf{glue}(a,-)}(\mathsf{glue}(b,c)) : \mathsf{glue}(a, \mathsf{inl}(b)) =^{P}_{\mathsf{glue}(b,c)} \mathsf{glue}(a, \mathsf{inr}(c)),$$

$$\mathsf{apd}_{\mathsf{glue}(a,-)}(\mathsf{glue}(b,c)) : \mathsf{transport}^{P}(\mathsf{glue}(b,c), \mathsf{glue}(a, \mathsf{inl}(b))) = \mathsf{glue}(a, \mathsf{inr}(c)),$$

and thus

$$\begin{aligned} \mathsf{glue}(a, \mathsf{inl}(b)) \cdot \mathsf{inr}(\mathsf{glue}(b,c)) &= \mathsf{transport}^{P}(\mathsf{glue}(b,c), \mathsf{glue}(a, \mathsf{inl}(b))) \\ &= \mathsf{glue}(a, \mathsf{inr}(c)), \end{aligned}$$

where the last path is $\mathsf{apd}_{\mathsf{glue}(a,-)}(\mathsf{glue}(b,c))$. □

The proof below differs from the proof in the book only by a few details.

Proof of Lemma 8.5.9. We define a map $f : (A*B)*C \to A*(B*C)$ by induction. We first need to define $f \circ \mathsf{inl} : A*B \to A*(B*C)$ and $f \circ \mathsf{inr} : C \to A*(B*C)$, which will be done by induction, and then

$$\mathsf{ap}_f \circ \mathsf{glue} : \prod_{t:(A*B)\times C} f(\mathsf{inl}(\mathsf{pr}_1(t))) = f(\mathsf{inr}(\mathsf{pr}_2(t))), \tag{62}$$

which will be done by induction on the first component of t.

First we define $f \circ \mathsf{inl} : A*B \to A*(B*C)$ and $f \circ \mathsf{inr} : C \to A*(B*C)$ by

$$\begin{aligned}(f \circ \mathsf{inl})(\mathsf{inl}(a))) &:\equiv \mathsf{inl}(a),\\ (f \circ \mathsf{inl})(\mathsf{inr}(b))) &:\equiv \mathsf{inr}(\mathsf{inl}(b)),\\ (f \circ \mathsf{inl})(\mathsf{glue}(a,b)) &:= \mathsf{glue}(a, \mathsf{inl}(b)),\\ f(\mathsf{inr}(c)) &:\equiv \mathsf{inr}(\mathsf{inr}(c)).\end{aligned} \tag{63}$$

We start the definition of (62).

Let $c : C$ and define $Q : A*B \to \mathcal{U}$ by

$$Q(x) :\equiv \Big(f(\mathsf{inl}(x)) =_{A*(B*C)} f(\mathsf{inr}(c))\Big),$$

that is,

$$Q(x) :\equiv \Big(f(\mathsf{inl}(x)) =_{A*(B*C)} \mathsf{inr}(\mathsf{inl}(b))\Big).$$

We first define

$$g : \prod_{x:A*B} Q(x). \tag{64}$$

Let $a : A$ and define $g_1(a) : Q(a)$, that is

$$g_1(a) : \mathsf{inl}(a) =_{A*(B*C)} \mathsf{inr}(\mathsf{inr}(c)),$$

by $g_1(a) :\equiv \mathsf{glue}(a, \mathsf{inr}(c))$. Let $b : B$ and define $g_2(b) : Q(b)$, that is

$$g_2(b) : \mathsf{inr}(\mathsf{inl}(b)) =_{A*(B*C)} \mathsf{inr}(\mathsf{inr}(c)),$$

by $g_2(b) :\equiv \mathsf{inr}(\mathsf{glue}(b,c))$. Let $a{:}A,\ b{:}B$ and define

$$g_3(a,b) : g_1(a) =^{Q}_{\mathsf{glue}(a,b)} g_2(b),$$

that is

$$g_3(a,b) : \mathsf{glue}(a, \mathsf{inr}(c)) =^{Q}_{\mathsf{glue}(a,b)} \mathsf{inr}(\mathsf{glue}(b,c)),$$

that is

$$g_3(a,b) : \mathsf{transport}^Q(\mathsf{glue}(a,b), \mathsf{glue}(a,\mathsf{inr}(c))) = \mathsf{inr}(\mathsf{glue}(b,c)),$$

as being the composite of the paths

$$\mathsf{transport}^Q(\mathsf{glue}(a,b), \mathsf{glue}(a,\mathsf{inr}(c))) = f(\mathsf{inl}(\mathsf{glue}(a,b)^{-1}))\bullet \mathsf{glue}(a,\mathsf{inr}(c))$$

$$= \mathsf{glue}(a,\mathsf{inl}(b))^{-1}\bullet \mathsf{glue}(a,\mathsf{inr}(c)) = \mathsf{inr}(\mathsf{glue}(b,c)),$$

where the last two paths are respectively given by (63) and the claim. Then we get a dependent function g as in (64) satisfying

$$g(\mathsf{inl}(a)) \equiv g_1(a), \quad g(\mathsf{inr}(b)) \equiv g_2(b), \quad \mathsf{apd}_g(\mathsf{glue}(a,b)) = g_3(a,b)$$

for all $a : A$ and all $b : B$.

We can set $f(\mathsf{glue}(x,c)) := g(x)$ for all $x : A * B$, so that $f(\mathsf{glue}(t))$ is now defined for all $t : (A * B) \times C$. This completes the definition of (62).

We get our desired map $f : (A * B) * C \to A * (B * C)$. Similarly, we can define a map $h : A * (B * C) \to (A * B) * C$, and checking that f and h are inverse to each other is a long and tedious but essentially straightforward computation. □

8.4 Theorem 8.5.11

The proof of Theorem 8.5.11 uses implicitly the 0-connectedness of the circle $\mathbb{S}^1$, 0-connectedness which has been observed in §7.15 p. 60 above.

8.5 Proof of Definition 8.6.5

The penultimate sentence of the proof of Definition 8.6.5 is

“To show that it is a family of equivalences, since being an equivalence is a mere proposition and $x_0 : 1 \to X$ is (at least) (-1)-connected, it suffices to assume x_1 is x_0.”

This follows from Lemma 8.6.1 (even from Lemma 7.5.7).

8.6 Lemma 8.6.10

Recall the statement:

Lemma 8.6.10. *Let $A : \mathcal{U}$, $B : A \to \mathcal{U}$, and $C : \prod_{a:A} B(a) \to \mathcal{U}$, and also $a_1, a_2 : A$ with $m : a_1 = a_2$ and $b : B(a_2)$. Then the function*

$$\mathsf{transport}^{\widehat{C}}(\mathsf{pair}^{=}_{B}(m,t), -) : C(a_1, \mathsf{transport}^{B}(m^{-1}, b)) \to C(a_2, b),$$

where $t : \mathsf{transport}^{B}(m, \mathsf{transport}^{B}(m^{-1}, b)) = b$ is the obvious coherence path and $\widehat{C} : (\sum_{a:A} B(a)) \to \mathcal{U}$ is the uncurried form of C, is equal to the equivalence obtained by univalence from the composite

$$\begin{aligned} C(a_1, \mathsf{transport}^{B}(m^{-1}, b)) &= \mathsf{transport}^{\lambda a.(B(a)\to\mathcal{U})}(m, C(a_1))(b) \\ &= C(a_2, b), \end{aligned}$$

where the paths are respectively given by (2.9.4) and $\mathsf{happly}(\mathsf{apd}_C(m), b)$.

(I have written $\mathsf{pair}^{=}_{B}$ instead of $\mathsf{pair}^{=}$ to stress the fact that this operation is taken with respect to the type family B.)

Here is a minor rephrasing:

If $p : C(a_1, \mathsf{transport}^{B}(m^{-1}, b)) = C(a_2, b)$ is the above path, then we have a path

$$\mathsf{transport}^{Z \mapsto Z}(p) = \mathsf{transport}^{\widehat{C}}(\mathsf{pair}^{=}_{B}(m,t)).$$

in $C(a_1, \mathsf{transport}^{B}(m^{-1}, b)) \to C(a_2, b)$.

8.7 Proof of Theorem 8.6.4

Theorem 8.6.4 is proved just before the statement of Corollary 8.6.14. The last display of the proof contains eight underscores; denote them by $u_1, \dots, u_8$. We read, after the display:

"…the underscore ought to be filled in with suitable coherence paths. Here the first step is functoriality of transport, the second invokes (8.6.12), and the third invokes (8.6.11) (with transport moved to the other side). Thus we have the same first component as the left-hand side of (8.6.13). We leave it to the reader to verify that the coherence paths all cancel, giving reflexivity in the second component."

To fill in the underscores, we introduce the following notation: If $p : y_1 = y_2$ and $q : N = y_1$ are paths in ΣX, we denote by

$$\alpha(p,q) : \mathsf{transport}^{N=-}(p,q) = q \cdot p$$

the obvious path in $N = y_2$. If $y_1 \equiv N$ and $q \equiv \mathsf{refl}_N$, we define

$$\beta(p) : \mathsf{transport}^{N=-}(p, \mathsf{refl}_N) = p$$

by $\beta(p) :\equiv \alpha(p, \mathsf{refl}_N) \bullet \omega$, where $\omega : \mathsf{refl}_N \bullet p = p$ is the obvious path in $N = y_2$. We also denote by

$$\gamma : \mathsf{merid}(x_0) \bullet \mathsf{merid}(x_0)^{-1} = \mathsf{refl}_N$$

the obvious path in $N = N$.

The underscores $u_1, \ldots, u_8$ can be chosen as follows:

$$u_1 :\equiv \beta(\mathsf{merid}(x_1) \bullet \mathsf{merid}(x_0)^{-1}), \quad u_2 :\equiv u_5 :\equiv \gamma,$$

$$u_3 :\equiv u_6 :\equiv \alpha(\mathsf{merid}(x_0)^{-1}, \mathsf{merid}(x_1)), \quad u_4 :\equiv \beta(\mathsf{merid}(x_1)),$$

$$u_7 :\equiv \mathsf{refl}_{\mathsf{merid}(x_1)}, \quad u_8 :\equiv \mathsf{refl}_{\mathsf{merid}(x_1) \bullet \mathsf{merid}(x_0)^{-1}}.$$

In other words, abbreviating $\mathsf{transport}^{\widehat{\mathsf{code}}}$ by t, merid by m, $\mathsf{pair}^=$ by p, refl by r, we get

$$\mathsf{t}\Big(\mathsf{p}\Big(\mathsf{m}(x_1) \bullet \mathsf{m}(x_0)^{-1}, \beta\Big(\mathsf{m}(x_1) \bullet \mathsf{m}(x_0)^{-1}\Big)\Big), \Big|(x_0, \gamma)\Big|_{2n}\Big)$$

$$= \mathsf{t}\Big(\mathsf{p}\Big(\mathsf{m}(x_0)^{-1}, \alpha\Big(\mathsf{m}(x_0)^{-1}, \mathsf{m}(x_1)\Big)\Big), \mathsf{t}\Big(\mathsf{p}\Big(\mathsf{m}(x_1), \beta\big(\mathsf{m}(x_1)\big)\Big), \Big|(x_0, \gamma)\Big|_{2n}\Big)\Big)$$

$$= \mathsf{t}\Big(\mathsf{p}\Big(\mathsf{m}(x_0)^{-1}, \alpha\Big(\mathsf{m}(x_0)^{-1}, \mathsf{m}(x_1)\Big)\Big), \Big|(x_1, \mathsf{r}_{\mathsf{m}(x_1)})\Big|_{2n}\Big)$$

$$= \big|(x_1, \mathsf{r}_{\mathsf{m}(x_1) \bullet \mathsf{m}(x_0)^{-1}})\big|_{2n}.$$

For the reader's convenience we spell out the following judgments:

$$\mathsf{code}(N, p) :\equiv \left\| \sum_{x:X} \Big(\mathsf{m}(x) \bullet \mathsf{m}(x_0)^{-1} = p\Big) \right\|_{2n}$$

for $p : N = N$,

$$|(x_0,\gamma)|_{2n} : \mathsf{code}(N,\mathsf{r}_N),$$

$$\mathsf{p}\Big(\mathsf{m}(x_1)\bullet\mathsf{m}(x_0)^{-1},\beta\Big(\mathsf{m}(x_1)\bullet\mathsf{m}(x_0)^{-1}\Big)\Big) : (N,\mathsf{r}_N) = (N,\mathsf{m}(x_1)\bullet\mathsf{m}(x_0)^{-1}),$$

$$\mathsf{p}\Big(\mathsf{m}(x_1),\beta\Big(\mathsf{m}(x_1)\Big)\Big) : (N,\mathsf{r}_N) = (S,\mathsf{m}(x_1)),$$

$$\mathsf{p}\Big(\mathsf{m}(x_0)^{-1},\alpha\Big(\mathsf{m}(x_0)^{-1},\mathsf{m}(x_1)\Big)\Big) : (S,\mathsf{m}(x_1)) = (N,\mathsf{m}(x_1)\bullet\mathsf{m}(x_0)^{-1}),$$

$$\mathsf{code}(S,q) :\equiv \left\|\sum_{x:X}\Big(\mathsf{m}(x)=q\Big)\right\|_{2n}$$

for $q : N = S$,

$$|(x_1,\mathsf{r}_{\mathsf{m}(x_1)}))|_{2n} : \mathsf{code}(S,\mathsf{m}(x_1)),$$

$$|(x_1,\mathsf{r}_{\mathsf{m}(x_1)\bullet\mathsf{m}(x_0)^{-1}}))|_{2n} : \mathsf{code}(N,\mathsf{m}(x_1)\bullet\mathsf{m}(x_0)^{-1}).$$

8.8 Proofs of Theorem 8.8.1 and Corollary 8.8.2

The assertions “hence $\|f(a)=b\|_{-1}$” in the proof of Theorem 8.8.1 and “hence $\|x=y\|_{-1}$” in the proof of Corollary 8.8.2 follow from Theorem 7.3.12.

8.9 Proof of Corollary 8.8.5

8.9.1 First part

The first part of the proof of Corollary 8.8.5 uses implicitly the following fact:

If C is a pointed type and $k,n:\mathbb{N}$ satisfy $0\le k\le n$, then we have

$$\pi_k(C)=\pi_k(\|C\|_n).$$

Proof. We have

$$\pi_k(C) = \Omega^k(\|C\|_k) = \Omega^k(\|\|C\|_n\|_k) = \pi_k(\|C\|_n),$$

the three equalities resulting respectively from Corollary 7.3.14, Lemma 7.3.15 and Corollary 7.3.14 again. □

Let us spell out the proof of the contractibility of $\|\mathsf{fib}_f(f(a))\|_n$. Lemma 7.3.15 yields

$$\|\|\mathsf{fib}_f(f(a))\|_n\|_0 = \|\mathsf{fib}_f(f(a))\|_0,$$

and we already know that $\|\mathsf{fib}_f(f(a))\|_0$ is contractible. As a result, $\|\mathsf{fib}_f(f(a))\|_n$ is 0-connected, and Corollary 8.8.4 implies that $\|\mathsf{fib}_f(f(a))\|_n$ is contractible. □

8.9.2 Second part

In the second part of the proof of Corollary 8.8.5, the assertion "therefore $\|b = f(a)\|_{-1}$" follows from Theorem 7.3.12.

8.10 Exercise 8.7

Statement. Define a **pointed equivalence** to be a pointed map whose underlying function is an equivalence.

1. Show that the type of pointed equivalences between pointed types (X, x_0) and (Y, y_0) is equivalent to $(X, x_0) =_{\mathcal{U}_\bullet} (Y, y_0)$.
2. Reformulate the notion of pointed equivalence in terms of a pointed quasi-inverse and pointed homotopies, in one of the coherent styles from Chapter 4.

Solution. It might help the reader to take a look at §2.4 p. 15 above. The first idea would be to define the type $(X, x_0) \simeq (Y, y_0)$ as being

$$\sum_{f:(X,x_0)\to(Y,y_0)} \mathsf{isequiv}(\mathsf{pr}_1(f)),$$

but we prefer the equivalent definition

$$((X, x_0) \simeq (Y, y_0)) :\equiv \sum_{f_1:X\to Y} \mathsf{isequiv}(f_1) \times (f_1(x_0) = y_0).$$

We shall define the maps a, b and c in the diagram

$$\begin{tikzcd}
\boxed{(X, x_0) = (Y, y_0)} \arrow[r, shift left, "a"] \arrow[d, "\mathsf{pr}_1"'] & \boxed{(X, x_0) \simeq (Y, y_0)} \arrow[l, shift left, "b"] \arrow[d, "c"] \\
\boxed{X = Y} & \boxed{X \simeq Y}
\end{tikzcd}$$

We define a by path induction in the obvious way. We define c by the formula

$$c(f_1, u, f_0) :\equiv (f_1, u).$$

We define b by setting

$$b(f_1, u, f_0) :\equiv \Big(\mathsf{ua}(f_1, u),\ q(f_1, u) \bullet f_0\Big),$$

where $q(f_1, u)$ is the obvious path from $p_{1\star}(x_0)$ to $f_1(x_0)$ in Y. We claim that a and b are inverse. We prove

$$b \circ a = \mathsf{id}_{(X,x_0)=(Y,y_0)}$$

by path induction. Let us check

$$a \circ b = \mathsf{id}_{(X,x_0)\simeq(Y,y_0)}.$$

Let $(f_1, u, f_0) : (X, x_0) \simeq (Y, y_0)$. We must show

$$a\Big(\mathsf{ua}(f_1, u),\ q(f_1, u) \bullet f_0\Big) = (f_1, u, f_0). \tag{65}$$

Defining $p_1 : X = Y$ by $p_1 :\equiv \mathsf{ua}(f_1, u)$, Equality (65) is equivalent to

$$a\bigg(p_1,\ q\Big(\mathsf{idtoeqv}(p_1)\Big) \bullet f_0\bigg) = (p_1, f_0),$$

which is easily proved by path induction on p_1.

9 Chapter 9

9.1 Theorem 9.2.5

In the penultimate paragraph of the proof of Theorem 9.2.5, one can also argue by path induction.

9.2 Lemma 9.2.8

To prove Lemma 9.3.2 (see §9.3 below) the following two lemmas might be useful. (Lemma 1 below is a mild generalization of Lemma 9.2.8.)

Lemma 1. *In the diagram below, A, B, C, D are precategories; $F, G : A \to B$, $H, K : C \to D$ and $L : B \to C$ are functors; $\gamma : F \to G$ and $\delta : H \to K$ are natural transformations:*

$$\begin{array}{ccccccc}
 & H & & & & F & \\
 & \leftarrow & & L & & \leftarrow & \\
D & \delta\downarrow & C & \leftarrow & B & \gamma\downarrow & A, \\
 & \leftarrow & & & & \leftarrow & \\
 & K & & & & G &
\end{array}$$

We have, in the above setting,

$$(\delta LG)(HL\gamma) = (KL\gamma)(\delta LF). \tag{66}$$

Proof. It suffices to check componentwise: at $a : A$ we have

$$\begin{aligned}
((\delta LG)(HL\gamma))_a &\equiv (\delta LG)_a(HL\gamma)_a \\
&\equiv \delta_{LGa} \circ HL(\gamma_a) \\
&= KL(\gamma_a) \circ \delta_{LFa} && \text{by naturality of } \delta \\
&\equiv (KL\gamma)_a \circ (\delta LF)_a \\
&\equiv ((KL\gamma)(\delta LF))_a. \ \square
\end{aligned}$$

Equality (66) can be illustrated by the commutative square

$$\begin{tikzcd}
HLF \arrow[r, "(HL\gamma)"] \arrow[d, "(\delta LF)"'] & HLG \arrow[d, "(\delta LG)"] \\
KLF \arrow[r, "(KL\gamma)"'] & KLG.
\end{tikzcd}$$

Lemma 2. *In the diagram below, A, B, C, D are precategories; $F : A \to B$, $G, H, K : B \to C$ and $L : C \to D$ are functors; $\alpha : G \to H$ and $\beta : H \to K$ are* *natural transformations:*

$$\begin{tikzcd} D & C \arrow[l, "L"'] & B \arrow[l, bend right=60, "K"', ""{name=K, below}] \arrow[l, "H"{description}, ""{name=H}] \arrow[l, bend left=60, "G", ""{name=G, above}] & A. \arrow[l, "F"'] \arrow[Rightarrow, from=G, to=H, "\alpha"'] \arrow[Rightarrow, from=H, to=K, "\beta"'] \end{tikzcd}$$

We have, in the above setting,

$$(L\beta F)(L\alpha F) = L(\beta\alpha)F. \tag{67}$$

Proof. It suffices to check componentwise: at $a : A$ we have

$$\begin{aligned} ((L\beta F)(L\alpha F))_a &\equiv (L\beta F)_a(L\alpha F)_a \\ &\equiv L(\beta_{Fa}) \circ L(\alpha_{Fa}) \\ &= L(\beta_{Fa} \circ \alpha_{Fa}) \\ &= L((\beta\alpha)_{Fa}) \\ &= (L(\beta\alpha)F)_a. \ \square \end{aligned}$$

9.3 Lemma 9.3.2

In the notation of §9.2 above, the diagrams and equalities below (which follow from (66) above) will help us verify the equality $\delta\gamma = 1_G$ in the proof of Lemma 9.3.2:

$$\begin{array}{ccccccc} & 1_A & & & & FG & \\ & \leftarrow & & G' & & \leftarrow & \\ A & \eta\downarrow & A & \leftarrow & B & \varepsilon\downarrow & B, \\ & \leftarrow & & & & \leftarrow & \\ & GF & & & & 1_B & \end{array}$$

$$(\eta\ G'\ 1_B)(1_A\ G'\ \varepsilon) = (GF\ G'\ \varepsilon)(\eta\ G'\ FG), \tag{68}$$

$$\begin{tikzcd} G'FG \arrow[r, "(1_A\ G'\ \varepsilon)"] \arrow[d, "(\eta\ G'\ FG)"'] & G' \arrow[d, "(\eta\ G'\ 1_B)"] \\ GFG'FG \arrow[r, "(GF\ G'\ \varepsilon)"'] & GFG', \end{tikzcd}$$

$$\begin{array}{ccccccc} & FG' & & & & FG & \\ & \leftarrow & & 1_B & & \leftarrow & \\ B & \varepsilon'\downarrow & B & \leftarrow & B & \varepsilon\downarrow & B, \\ & \leftarrow & & & & \leftarrow & \\ & 1_B & & & & 1_B & \end{array}$$

$$(\varepsilon'\ 1_B\ 1_B)(FG'\ 1_B\ \varepsilon) = (1_B\ 1_B\ \varepsilon)(\varepsilon'\ 1_B\ FG), \tag{69}$$

$$\begin{tikzcd} FG'FG \arrow[r, "(FG'\ 1_B\ \varepsilon)"] \arrow[d, "(\varepsilon'\ 1_B\ FG)"'] & FG' \arrow[d, "(\varepsilon'\ 1_B\ 1_B)"] \\ FG \arrow[r, "(1_B\ 1_B\ \varepsilon)"'] & 1_B, \end{tikzcd}$$

$$
\begin{array}{ccccccc}
 & 1_A & & & & 1_A & \\
 & \leftarrow & & 1_A & & \leftarrow & \\
A & \eta\downarrow & A & \leftarrow & A & \eta'\downarrow & A, \\
 & \leftarrow & & & & \leftarrow & \\
 & GF & & & & G'F &
\end{array}
$$

$$(\eta\ 1_A\ G'F)(1_A\ 1_A\ \eta') = (GF\ 1_A\ \eta')(\eta\ 1_A\ 1_A), \tag{70}$$

$$\begin{tikzcd}
1_A \arrow[r, "(1_A\ 1_A\ \eta')"] \arrow[d, "(\eta\ 1_A\ 1_A)"'] & G'F \arrow[d, "(\eta\ 1_A\ G'F)"] \\
GF \arrow[r, "(GF\ 1_A\ \eta')"'] & GFG'F.
\end{tikzcd}$$

Here is the verification of the equality $\delta\gamma = 1_G$ in the proof of Lemma 9.3.2 announced at the beginning of this section:

$$
\begin{aligned}
\delta\gamma &\equiv (G\varepsilon')(\eta G')(G'\varepsilon)(\eta' G) & \\
&= (G\varepsilon')(GFG'\varepsilon)(\eta G'FG)(\eta' G) & \text{by (68)} \\
&= G\,(\varepsilon'\,(FG'\varepsilon))\,((\eta G'F)\,\eta')\,G & \text{by (67)} \\
&= G\,(\varepsilon\,(\varepsilon' FG))\,((GF\eta')\,\eta)\,G & \text{by (69) and (70)} \\
&= G\,(\varepsilon\,(\varepsilon' F))\,((F\eta')\,\eta)\,G & \text{by (67)} \\
&= (G\varepsilon)(\varepsilon' F)(F\eta')(\eta G) & \\
&= (G\varepsilon)(\eta G) & \text{by } \Delta_1 \\
&= 1_G & \text{by } \Delta_2,
\end{aligned}
$$

where Δ_1 and Δ_2 denote the first and second triangle identity.

9.4 Proof of Lemma 9.4.9

• The following observation, which could be made just before Definition 9.2.3, is used in the proof of Lemma 9.4.9.

In the setting of Definition 9.2.1, let $p : F_0 = G_0$*; for each* $a : A_0$ *let* $p_a : Fa = Ga$ *be the induced path; let* $a, a' : A_0$*; and define* $P : (A_0 \to B_0) \to \mathcal{U}$ *by* $P(f) :\equiv (\hom_A(a, a') \to \hom_B(fa, fa'))$*. Then we have*

$$p_\star(F_{a,a'})(g) = \mathsf{idtoiso}(p_{a'}) \circ F_{a,a'}(g) \circ \mathsf{idtoiso}((p_a)^{-1})$$

for all $g : \hom_A(a, a')$*.*

• In the penultimate paragraph of the proof of Lemma 9.4.9, the phrase "the *other* triangle identity, which we have seen in Chapter 4 is equivalent to (9.4.11)" refers (I think) to Lemma 4.2.2.

10 Chapter 10

10.1 A Lemma

Recall the notion of image (Definition 7.6.3 of the book):

Definition 7.6.3. Let $f : A \to B$ be a function. The n**-image** of f is defined as

$$\mathsf{im}_n(f) :\equiv \sum_{b:B} \|\mathsf{fib}_f(b)\|_n.$$

When $n = -1$, we write simply $\mathsf{im}(f)$ and call it the **image** of f.

The following lemma, whose proof is straightforward, will be used in §10.2 below.

Lemma. *Let* $f : A \to B$ *be a map. Assume that* B *is a set. Then* $\mathsf{im}(f)$ *is a mere proposition if and only if* $f(a) = f(a')$ *for all* $a, a' : A$*.*

Proof. Suppose $\mathsf{im}(f)$ is a mere proposition and let $a, a' : A$. As

$$\Big(f(a), |(a, \mathsf{refl}_{f(a)})|\Big), \Big(f(a'), |(a', \mathsf{refl}_{f(a')})|\Big) : \mathsf{im}(f),$$

we have $f(a) = f(a')$. Conversely, assume $f(a) = f(a')$ for all $a, a' : A$. Let $x, x' : \mathsf{im}(f)$. We claim $x = x'$. We may suppose $x \equiv (b, u), x' \equiv (b', u')$ (obvious notation). It suffices to show $b = b'$. As $b = b'$ is a mere proposition, we may assume $u \equiv |(a, p)|$ and $u' \equiv |(a', p'))|$ with

$$a, a' : A, \quad p : f(a) = b, \quad p' : f(a') = b'.$$

This implies $b = b'$, as required. □

10.2 Before Lemma 10.1.1

This is about the sentence "The obvious candidate for the coequalizer of the kernel pair of $f : A \to B$ is the image of f, as defined in §7.6" a few paragraphs before

Lemma 10.1.1. (It is clear from the context that A and B are sets.) Let us show that this obvious candidate is the good one. By Lemma 4.8.2 of the book (see §4.6 p. 35) it suffices to prove the following statement:

If $B : A \to \mathsf{Set}$ is a set family over a set A and $\sim$ is the equivalence relation on $\sum_{a:A} B(a)$ defined by

$$(a, b) \sim (a', b') \textit{ if and only if } a = a',$$

then the resulting quotient satisfies

$$\left(\sum_{a:A} B(a)\right)\Big/\!\sim \quad = \quad \sum_{a:A} \|B(a)\|.$$

Proof. Let us define a map

$$h : \left(\sum_{a:A} \|B(a)\|\right) \quad \to \quad \left(\sum_{a:A} B(a)\right)\Big/\!\sim .$$

It suffices to define, keeping $a_0 : A$ fixed, the object $h(a_0, \beta)$ for all $\beta : \|B(a_0)\|$. Define

$$f : B(a_0) \to \left(\sum_{a:A} B(a)\right)\Big/\!\sim$$

by $f(b) :\equiv q(a_0, b)$ for all $b : B(a_0)$, and let

$$\tilde{f} : B(a_0) \to \mathsf{im}(f), \quad \mathsf{pr}_1 : \mathsf{im}(f) \to \left(\sum_{a:A} B(a)\right)\Big/\!\sim$$

be the canonical factorization of f (see Lemma 7.6.4 of the book). The lemma in §10.1 above implies that $\mathsf{im}(f)$ is a mere proposition. As a result, $\tilde{f}$ induces a map $g : \|B(a_0)\| \to \mathsf{im}(f)$ satisfying $g(|b|) \equiv f(b)$ for all $b : B(a_0)$, and we can set $h(a_0, \beta) :\equiv \mathsf{pr}_1(g(\beta))$ for all $b : B(a_0)$.

We leave it to the reader to check, using §7.5 above, that h is invertible. □

10.3 Proof of Lemma 10.1.13

To prove the equality $\mathsf{merid}_\star(\tau(N)) = \tau(S)$, one can use the following lemma, whose proof is straightforward:

Lemma. *Let $P : A \to B \to \mathcal{U}$ be a type family, define $Q : A \to \mathcal{U}$ by*

$$Q(a) :\equiv \prod_{b:B} P(a,b),$$

let $p : a = a'$ be a path in A, let $f : Q(a)$, and let $b : B$. Then we have

$$\mathsf{transport}^{Q}(p,f)(b) = \mathsf{transport}^{P(-,b)}(p, f(b)).$$

10.4 The induction principle for acc (§10.3)

Here is a first particular case of the general induction principle for acc:

If we have a type family $P : A \to \mathcal{U}$ and

$$f : \prod_{a:A} \left(\prod_{b:A} (b < a) \to \mathsf{acc}(b) \times P(b) \right) \to P(a),$$

then we get

$$g : \prod_{a:A} \mathsf{acc}(a) \to P(a)$$

with

$$g\big(a, \mathsf{acc}_{<}(a, \mathsf{pr}_1 \circ u)\big) \equiv f\bigg(a \ , \ \lambda b.\lambda l.\Big(\mathsf{pr}_1 u(b,l) \ , \ g\big(b, \mathsf{pr}_1 u(b,l)\big)\Big)\bigg)$$

for all $a : A$ and all $u : \prod_{b:A} (b < a) \to \mathsf{acc}(b) \times P(b)$.

Here is a second particular case:

If $<$ is well-founded, if we have a type family $P : A \to \mathcal{U}$, and if we have

$$f : \prod_{a:A} \left(\prod_{b:A} (b < a) \to P(b) \right) \to P(a), \tag{71}$$

then we get $g : \prod_{a:A} P(a)$ with $g(a) \equiv f(a, \lambda b.\lambda l.g(b))$ for all $a : A$.

(We are using Lemma 10.3.2 of the book.)

10.5 Lemma 10.3.7

Here is a minor variant:

Suppose P is a set and we have a function $h : (P \to \mathsf{Prop}) \to P$. Then if $<$ is a well-founded relation on A, there is a function $g : A \to P$ such that for all $a : A$ we have

$$g(a) \equiv h\left(\{g(b) \mid b < a\}\right),$$

where $\{g(b) \mid b < a\} : P \to \mathsf{Prop}$ maps $p : P$ to

$$\exists\ (b : A)\ .\ (g(b) = p)\ \wedge\ (b < a).$$

Proof. We shall use the induction principle containing Display (71) above. Define

$$f : \prod_{a:A} \left(\prod_{b:A} (b < a) \to P \right) \to P$$

by

$$f(a, s) :\equiv h(\{s(b, l) \mid l : b < a\}),$$

where $\{s(b, l) \mid l : b < a\} : P \to \mathsf{Prop}$ maps $p : P$ to

$$\exists\ (b : A)\ .\ \exists\ (l : b < a)\ .\ s(b, l) = p.$$

We get

$$g(a) \equiv f(a, \lambda b.\lambda l.g(b)) \equiv h(\{g(b) \mid b < a\}). \ \square$$

10.6 Lemma 10.3.8

Recall the statement:

Assuming excluded middle, $<$ is well-founded if and only if every nonempty subset $B : \mathcal{P}(A)$ merely has a minimal element.

Here is a slightly different phrasing of the proof.

Proof. Suppose first $<$ is well-founded and A has no minimal element. That is, for any $a : A$ there merely exists a $b : A$ with $b < a$. We shall show $A = \mathbf{0}$. We first claim

$$\prod_{a:A} \left(\prod_{a':A} (a' < a) \to \mathbf{0} \right) \to \mathbf{0}. \tag{72}$$

Let $a : A$ and $s : \prod_{a':A}(a' < a) \to \mathbf{0}$. It suffices to prove $\mathbf{0}$. As observed above, there is a $b : A$ and an $l : b < a$, yielding $s(b, l) : \mathbf{0}$. This proves (72). Then $\prod_{a:A} \mathbf{0}$, that is $A \to \mathbf{0}$, follows by well-founded induction, and implies $A = \mathbf{0}$.

Now suppose that each nonempty subset merely has a minimal element, and assume that an inaccessible $a_0 : A$ merely exists. As it suffices to prove $\mathbf{0}$, we can assume that a_0 exists purely, and that it is minimal for the property of being inaccessible, which means that we have an $h : \prod_{b:A}(b < a_0) \to \mathsf{acc}(b)$, yielding $\mathsf{acc}_<(a_0, h) : \mathbf{0}$, as required. □

10.7 Lemma 10.3.12

Recall the statement: *Any simulation is injective.*

Here is a slightly different phrasing of the proof:

Proof. We prove by well-founded induction on $a : A$ that for any $b : A$, if $f(a) = f(b)$ then $a = b$. The inductive hypothesis for $a : A$ says that for any $a' < a$, and any $b : A$, if $f(a') = f(b)$ then $a' = b$.

Suppose $f(a) = f(b)$; we must show $a = b$. By extensionality, it suffices to show that for any $c : A$ we have $(c < a) \Leftrightarrow (c < b)$.

If $c < a$, then $f(c) < f(a)$ by Definition 10.3.11(i) (see §3.4 p. 26). Hence $f(c) < f(b)$, so by Definition 10.3.11(ii) there merely exists $c' : A$ with $c' < b$ and $f(c) = f(c')$. By the inductive hypothesis for a, we have $c = c'$, hence $c < b$.

If $c < b$, then $f(c) < f(b)$ by Definition 10.3.11(i). Hence $f(c) < f(a)$, so by Definition 10.3.11(ii) there merely exists $c' : A$ with $c' < a$ and $f(c) = f(c')$. By the inductive hypothesis for a, we have $c = c'$, hence $c < a$. □

10.8 Theorem 10.4.4

In the proof of Theorem 10.4.4 the notation $\{f(x) \mid P(x)\}$, introduced after Lemma 10.1.6, is used a lot.

11 Chapter 11

11.1 The definition of $\mathbb{Q}$ (§11.1)

The type $\mathbb{Q}$ is defined in §11.1 as $\mathbb{Q} :\equiv (\mathbb{Z} \times \mathbb{N})/\approx$, where $\approx$ is defined by

$$(u, a) \approx (v, b) :\equiv (u(b + 1) = v(a + 1)).$$

One may add something like "We leave it to the reader to check that $\approx$ is an equivalence relation".

11.2 §11.2.1

The following observation could be made just before Lemma 11.2.2:

For $x, y : \mathbb{R}_d$ the inequality $x < y$ implies $x \neq y$. Moreover the map $f : \mathbb{Q} \to \mathbb{R}_d$, $q \mapsto (L_q, U_q)$ satisfies

$$f(q) < f(r) \iff q < r \quad \text{and} \quad f(q) \leq f(r) \iff q \leq r.$$

11.3 Corollary to Lemma 11.2.2

If $q, r : \mathbb{Q}$ and $q < r$ and $x : \mathbb{R}_d$, then we merely have $q < x$ or $x < r$ by locatedness.

11.4 Proof of Theorem 11.2.12

The claim "Since $q + 2\varepsilon < r - 2\varepsilon$ merely $L_{x_\varepsilon}(q + 2\varepsilon)$ or $U_{x_\varepsilon}(q - 2\varepsilon)$" follows from §11.3 above.

12 Appendix A

It seems to me that the expression $\lambda x.b$ is defined in two conflicting ways, firstly at the beginning of A.1, and secondly in the last sentence of A.2.4.