Draft

Dimitrios Koutsoulis
11838639

August 17, 2019

Chapter 1

Type Theory

1.1 Introduction

Type theory is a formal language and deductive system, that is self sufficient in the sense that it need not be formulated as a collection of axioms on top of some other formal system like First Order Logic, instead its deductive system can be built on top of its own formal language. longer sentence to check formating

Central to Type Theory is the notion of *Type*. Every term a in Type Theory we come across, must lie in some type A, which we denote as $a : A$. Note that the relation $:$ is transitive, so $a : A$ and $A : B$ imply $a : B$.

For the deductive part of Type Theory, we interpret propositions as types. Proving proposition P amounts to providing some inhabitant $p : P$.

1.2 Type Construction Operations

Let's have a look at some important type constructions.

- Given types $A, B : \mathcal{U}$ we can define the type $A \to B$ of functions from A to B. We can use λ-abstraction to construct elements of this type. $\lambda x.\Phi$ lies in $A \to B$ iff for $a : A$ we have $\Phi[a/x] : B$. For $f : A \to B$ and $a : A$ we have that the application of f on a, denoted as $f(a)$ or $f\ a$, lies in B.

- Given some type $A : \mathcal{U}$ and a family of types B over A, $B : A \to \mathcal{U}$, we have the type of dependent functions

$$\prod_{a:A} B(a)$$

where for $f : \prod_{a:A} B(a)$ and $x : A$ we have $f(x) : B(x)$. As in the case of non-dependent functions, we can use lambda abstraction to construct elements of a dependently-typed function type.

- Given $A, B : \mathcal{U}$ we can define the product type $A \times B : \mathcal{U}$. For $a : A$ and $b : B$ we have the pair $(a, b) : A \times B$. We also have the projection functions

$$\mathtt{pr}_1 : A \times B \to A : (a, b) \mapsto a$$
$$\mathtt{pr}_2 : A \times B \to B : (a, b) \mapsto b$$

- Given $A : \mathcal{U}$ and family of types B over A, $B : A \to \mathcal{U}$, we can define the dependent pair type

$$\sum_{a:A} B(a)$$

Given $x : A$ and $b : B(x)$ we can construct the pair $(x, b) : \sum_{a:A} B(a)$. We have two projection functions, similar to the case of the product type.

- Given $A, B : \mathcal{U}$ we can construct the coproduct type $A + B$. We can construct elements of $A + B$ using the functions

$$\mathtt{inl} : A \to A + B$$
$$\mathtt{inr} : B \to A + B$$

This induces the induction principle

$$\mathtt{ind}_{A+B} : \prod_{C:A+B\to U} \Big(\prod_{a:A} C(\mathtt{inl}\ a)\Big) \to \Big(\prod_{b:A} C(\mathtt{inr}\ b)\Big) \to \prod_{x:A+B} C(x)$$

- Given $x, y : A$ we have the **identity type** $x =_A y$. An element of this type amounts to a proof that x and y are equal. Say x and y are judgmentally equal. This is captured by the element $\mathtt{idp}_x : x =_A y$.

The relevant induction principle describes how we can use elements of an identity type

$$\mathtt{ind}_{=_A} : \prod_{C:\prod_{x,y:A}(x=_A y)\to\mathcal{U}} \Big(\prod_{x:A} C(x,x,\mathtt{idp}_x)\Big) \to \prod_{x,y:A}\prod_{(p:x=_A y)} C(x,y,p)$$

The relevant computation gives us the judgmental equality

$$\mathtt{ind}_{=_A}(C,c,x,x,\mathtt{idp_x}) \equiv c\ x$$

We can concatenate those paths whose domains and codomains allow for it. Paths are equivalences. That is if $p : x = y$ is such a path, we can provide its inverse p^{-1} for which we have in turn a path between $p \cdot p^{-1}$ and $\mathtt{idp}_y$ and another one between $p^{-1} \cdot p$ and $\mathtt{idp}_x$.

- For every type A there is its **propositional truncation** $\|A\|$. For every element $a : A$ there exists $|a| : \|A\|$. For every $x, y : \|A\|$ we have $x = y$. Given mere proposition B and $f : A \to B$, the recursion principle gives us $g : \|A\| \to B$ such that $g(|a|) \equiv f(a)$ for all $a : A$.

1.3 Important types

Definition 1.3.1. *We call a type A a **mere proposition** if for every $a, b : A$ we have $a = b$.*

Definition 1.3.2. *We call a type A a **set** if for every $a, b : A$ we have that $a =_A b$ is a mere proposition.*

Definition 1.3.3. *We call a type A **contractible** if there exists $a : A$ such that for all $x : A$ it holds that $x = a$.*

1.4 Logic

Our informal deductions in Type Theory will be reminiscent of First Order Logic ones. To be able to use a similar verbiage, we will set down a handful of types, corresponding to the connectives that let us form well-formed formulas in FOL. These types need to be mere propositions, so that we can form non-constructive deductions. This approach is called 'propositions as h-propositions' in the HoTT book. In the following, A and B are mere propositions.

- When we talk of conjunction $A \wedge B$, we mean the product $A \times B$.
- We interpret $A \vee B$, as the truncation $\|A + B\|$.
- We interpret $\forall a \in A,\ P(a)$, where $P(a)$ is a mere proposition for all $a \in A$, as $\prod_{a:A} P(a)$.
- We interpret $\exists a \in A,\ P(a)$, where $P(a)$ is a mere proposition for all $a \in A$, as $\|\Sigma_{a:A} P(a)\|$

The use of the above notation is interchangeable in the sections to follow.

Chapter 2

Modalities

Definition 2.0.1. *A **modality** is a function $\bigcirc : \mathcal{U} \to \mathcal{U}$ with the following properties.*

1. *For every type A we have a function $\eta_A^{\bigcirc} : A \to \bigcirc A$*
2. *for every $A : \mathcal{U}$ and every type family $B : \bigcirc A \to \mathcal{U}$ we have a function*
$$\mathtt{ind}_{\bigcirc} : \Big(\prod_{a:A} \bigcirc(B(\eta_A^{\bigcirc}\ a))\Big) \to \prod_{z:\bigcirc A} \bigcirc(B\ z)$$
3. *For every $f : \prod_{a:A} \bigcirc(B(\eta_A^{\bigcirc}\ a))$ there is a path $\mathtt{ind}_{\bigcirc}(f)(\eta_A^{\bigcirc}\ a) = f\ a$*
4. *For all $z, z' : \bigcirc A$, the function $\eta_{z=z'}^{\bigcirc} : (z = z') \to \bigcirc(z = z')$ is an equivalence.*

A modality $\bigcirc$ induces a Σ-closed reflective subuniverse.

Definition 2.0.2. *Given modality $\bigcirc : \mathcal{U} \to \mathcal{U}$, a **reflective subuniverse** is a 'subset' of $\mathcal{U}$ encoded by a family of h-propositions $P : \mathcal{U} \to \mathtt{Prop}$ such that the following conditions hold.*

- *For $A : \mathcal{U}$, we have $P(\bigcirc A)$.*
- *For $A : \mathcal{U}$ and B such that $P(B)$, the function*
$$\lambda f. f \circ \eta_A^{\bigcirc} : (\bigcirc A \to B) \to (A \to B)$$
is an equivalence.

The subuniverse is Σ***-closed*** *if for* X *such that* $P(X)$ *and* $Q : X \to \mathcal{U}$ *such that* $\prod_{x:X} P(Q(x))$*, we have* $P(\Sigma_{x:X} Q(x))$*.*

Theorem 2.0.3. *Reflective subuniverses are closed under products. That is for subuniverse* P *and* $B : A \to \mathcal{U}$ *such that* $\prod_{a:A} P(B(a))$*, we have that* $P(\prod_{a:A} B(a))$*.*

Proof. For $a : A$, consider $\mathtt{ev}_a : (\prod_{a:A} B(a)) \to B(a)$ defined by $\mathtt{ev}_a(f) :\equiv f(x)$. Since $P(B(a))$, we have

$$(\lambda f. f \circ \eta^{\bigcirc}_{\prod_{a:A} B(a)})^{-1}(\mathtt{ev}_a) : \bigcirc\big(\prod_{a:A} B(a)\big) \to B(a)$$

We can now define the retraction of $\eta^{\bigcirc}_{\prod_{a:A} B(a)}$ by pattern matching as such: For $z : (\prod_{a:A} B(a))$ and $a : A$ we have

$$(\lambda f. f \circ \eta^{\bigcirc}_{\prod_{a:A} B(a)})^{-1}(\mathtt{ev}_a)(z) : B(a)$$

□

Definition 2.0.4. *For* $B : A \to \mathcal{U}$*, we call* X B***-null*** *if the map*

$$\lambda x. \lambda b. x : X \to (B(a) \to X)$$

is an equivalence for all $a : A$*.*

Chapter 3

Computability

In this chapter we will give an overview of basic definitions and results of Recursion theory and formalize them in Type theory. The reader is expected to be somewhat familiar with Turing machines, as we will not be delving into their technical details.

Definition 3.0.1. *A* ***partial function*** *is a function with a subset of the naturals as its domain.*
Formally

$$f : \Big(\sum_{n:\mathbb{N}} P\ n\Big) \to \mathbb{N}$$

where $P : \mathbb{N} \to \mathbf{Prop}$ *is a family of propositions over* $\mathbb{N}$*, which works as the characteristic function of the domain of the partial function* f*.*

A standard result in Recursion theory is that there exists a Turing machine that enumerates all Turing Machines. We assume a fixed enumeration $T_1, T_2, \ldots$ that we refer to going forward.

Definition 3.0.2. *Church's thesis* (CT) *is an axiom to be assumed, which states that every function* $\mathbb{N} \to \mathbb{N}$ *is computable. Formally in Type theory*

$$\prod_{f:\mathbb{N}\to\mathbb{N}} \Big\| \sum_{e:\mathbb{N}} \prod_{x:\mathbb{N}} \sum_{z:\mathbb{N}} T(e, x, z) \times U(z) = f(x) \Big\|$$

where T *is Kleene's predicate,* e *identifies a Turing machine,* x *is some input to* f *and its corresponding Turing machine* T_e*, finally* z *is the computation* *history that T_e goes through when given x as the input, with $U(z)$ being the output at the end of the computation.*

In other words, Church's thesis assures us that for every $f : \mathbb{N} \to \mathbb{N}$ there exists some computable function that agrees with it on every input. This holds true for partial functions too, since any one of them can be trivially extended to a total ($\mathbb{N} \to \mathbb{N}$) one for the purposes of Church's thesis.

Chapter 4

LLPO

Definition 4.0.1. *The Lesser Limited Principle of Omniscience, states that given binary sequence $s : \mathbb{N} \to \mathbf{2}$ and the fact that there is at most one occurence of* 1 *in the sequence, formally*

$$\texttt{atMost1one}\, s : \prod_{n_1:\mathbb{N}} \prod_{n_2:\mathbb{N}} s(n_1) = 1 \to s(n_2) = 1 \to n_1 = n_2$$

we can then have by the LLPO a witness for $\mathtt{p_{odd}} \vee \mathtt{p_{even}}$, where $\mathtt{p_{odd}}$ (with s as an implicit argument) is the statement that for all odd positions n, $s(n) = 0$, formally $\mathtt{p_{odd}} \equiv \prod_{n:\mathbb{N}} \mathtt{odd}(n) \to s(n) = 0$. Similarly for $\mathtt{p_{even}}$.

LLPO can be viewed as a weaker form of the Law of Excluded Middle.

Lemma 4.0.2. *The Law of Excluded Middle implies the Lesser Limited Principle of Omniscience.*

Proof. By LEM we have a witness $l_1 : \mathtt{p_{odd}} \vee \neg \mathtt{p_{odd}}$. Since this is a coproduct, by the relevant principle of induction, it's enough to prove LLPO from the disjuncts.

- $\mathtt{p_{odd}} \Rightarrow$ LLPO, trivially.
- $\neg \mathtt{p_{odd}}$, alongside LEM, implies that there exists odd $n_e : \mathbb{N}$ such that $s(n_e) = 1$. We can now prove $\prod_{n:\mathbb{N}} \mathtt{even}(n) \to s(n) = 0$. Let $n : \mathbb{N}$ such that $\mathtt{even}(n)$ is true. By the definition of s, $s(n) = 0 \vee s(n) = 1$. We invoke the principle of induction of coproducts and prove LLPO from the disjuncts.

- $s(n) = 0$, in this case we are done.
- $s(n) = 1$. By `atMost1one` we have $n = n_e \Rightarrow n$ even and odd which is a contradiction $c : \bot$. Ex Falso, `efq` $: \bot \to s(n) = 0$. Then $\texttt{efq}(c) : s(n) = 0$.

□

Lemma 4.0.3. *If we replace the consequent of LLPO with its double negation, let's call it LLPO$^{\neg\neg}$, then we can prove it in Type Theory.*

Proof. From LEM $\Rightarrow$ LLPO we have LEM $\Rightarrow$LLPO$^{\neg\neg}$ by effectively the same proof. Since we invoke LEM only twice, we can propose the double negation of these two instances of LEM where the need arises, show that they are provable in our context and then drop the double negation since it's a modality and the goal is of the same modality. We effectively use the same method as when we drop truncations around hypotheses when proving mere propositions, only this time we are working with the double negation modality.

The first instance we come across is

$$\mathsf{p}_{\mathsf{odd}} \vee \neg \mathsf{p}_{\mathsf{odd}}$$

We want to prove $\neg\neg(\mathsf{p}_{\mathsf{odd}} \vee \neg \mathsf{p}_{\mathsf{odd}})$. Assume $q : (\mathsf{p}_{\mathsf{odd}} \vee \neg \mathsf{p}_{\mathsf{odd}}) \to \bot$. We compose q with $|\cdot|$ to get

$$q' : (\mathsf{p}_{\mathsf{odd}} + \neg \mathsf{p}_{\mathsf{odd}}) \to \bot$$

We then have

$$q' \circ \texttt{inl} : \neg \mathsf{p}_{\mathsf{odd}}$$

and

$$q' \circ \texttt{inr} : \neg\neg \mathsf{p}_{\mathsf{odd}}$$

which lead us to falsum.

The second instance is

$$\left\| \sum_{n_e : \texttt{Odd}} s(n_e) = 1 \right\| \vee \neg \left\| \sum_{n_e : \texttt{Odd}} s(n_e) = 1 \right\|$$

Assume the negation of the above

$$q : \neg\Big(\Big\| \sum_{n_e:\texttt{Odd}} s(n_e) = 1\Big\| \vee \neg\Big\| \sum_{n_e:\texttt{Odd}} s(n_e) = 1\Big\|\Big)$$

towards contradiction. We compose with $|\cdot|$ to rid ourselves of the truncation

$$q' : \neg\Big(\Big\| \sum_{n_e:\texttt{Odd}} s(n_e) = 1\Big\| + \neg\Big\| \sum_{n_e:\texttt{Odd}} s(n_e) = 1\Big\|\Big)$$

We then have

$$q' \circ \texttt{inr} : \neg\neg\Big\| \sum_{n_e:\texttt{Odd}} s(n_e) = 1\Big\|$$

and

$$q' \circ \texttt{inl} : \neg\Big\| \sum_{n_e:\texttt{Odd}} s(n_e) = 1\Big\|$$

Contradiction. □

Definition 4.0.4. *The Axiom of Countable Choice (ACC) is adapted from its set-theoretic counterpart and states that if for every $n : \mathbb{N}$ there exists (merely) some $b : \|B\ n\|$, where $B : \mathbb{N} \to \mathcal{U}$ is a family of h-sets over $\mathbb{N}$, then there merely exists some $f : \prod_{n:\mathbb{N}} B\ n$.*
More concisely

$$\Big(\prod_{n:\mathbb{N}} \|B\ n\|\Big) \to \Big\| \prod_{n:\mathbb{N}} B\ n \Big\|$$

Lemma 4.0.5. *Under Church's thesis, the following type is inhabited*

$$\sum_{F:\mathbb{N}\to\mathbb{N}\to\mathbf{2}} \Big(\big(\prod_{m:\mathbb{N}} \texttt{atMost1one}\ F(m)\big) \times \big(\prod_{f:\mathbb{N}\to\mathbf{2}} \| \sum_{m,k:\mathbb{N}} F(m, 2k + f\ m) = 1\|\big)\Big)$$

Proof. First we provide a witness

$$G : \sum_{F:\mathbb{N}\to\mathbb{N}\to\mathbf{2}} \prod_{n:\mathbb{N}} \texttt{atMost1one}\ F(n)$$

For $n, m : \mathbb{N}$ we pick the indexed T_n Turing machine of our enumeration. We can decide whether m is odd or even.

- If it's odd, then there exists actual $k : \mathbb{N}$ such that $m = 2 * k + 1$. If k is the Gödel number of the computation history that T_n goes through when given n as the input, furthermore, if it halts at the end of this computation with output 1 then set $G.\mathtt{fst}\ n\ m :\equiv 1$. Otherwise set $G.\mathtt{fst}\ n\ m :\equiv 0$.
- If it's even, then there exists actual k such that $m = 2{*}k$. We work as in the odd case, with the only difference being that we set $G.\mathtt{fst}\ n\ m :\equiv 1$ iff the output of the halting computation is 0.

Having defined $G.\mathtt{fst}$ it's easy to see that $G.\mathtt{snd}$ has a straightforward proof, which we ommit.

We now define

$$Q : (\mathbb{N} \to \mathbb{N} \to \mathbf{2}) \to \mathcal{U}$$

by

$$Q\ F :\equiv \prod_{f:\mathbb{N}\to\mathbf{2}} \Big\| \sum_{m,k:\mathbb{N}} F(m, 2k + f\ m) = 1 \Big\|$$

We want to prove $Q\ G.\mathtt{fst}$. Consider arbitrary $f : \mathbb{N} \to \mathbf{2}$. By CT we have that there exists, merely, $e : \mathbb{N}$ such that T_e computes f, furthermore there exists $z : \mathbb{N}$ which z is the Gödel number of the halting computation that T_e goes through when given e as an input and lastly T_e halts at the end of this computation and outputs $j : \mathbb{N}$ where $j = f\ e$. Since it's decidable whether j is 0 or 1, we can take cases on it. In both cases we have $G.\mathtt{fst}(e, 2z + j) = 1$.

We conclude the proof by providing $G.\mathtt{fst}$ as the first component and the product of $G.\mathtt{snd}$ with $Q\ G.\mathtt{fst}$ as the second. □

Theorem 4.0.6.

$$ACC \times CT \times LLPO \to \bot$$

Proof. Let G be the witness we reached in our proof of Lemma 4.0.5. Let $F :\equiv G.\mathtt{fst}$. By LLPO we can procure

$$f : \prod_{n:\mathbb{N}} \|\mathsf{p}_{\mathtt{odd}}(F\ n) + \mathsf{p}_{\mathtt{even}}(F\ n)\|$$

By the ACC we get

$$f' : \Big\| \prod_{n:\mathbb{N}} \mathsf{p}_{\mathtt{odd}}(F\ n) + \mathsf{p}_{\mathtt{even}}(F\ n) \Big\|$$

Since our goal is $\bot$, which is an h-prop, we can ignore the truncation and act as if we have access to

$$f'' : \prod_{n:\mathbb{N}} \mathtt{p}_{\mathtt{odd}}(F\ n) + \mathtt{p}_{\mathtt{even}}(F\ n)$$

Let $e : \big(\mathtt{p}_{\mathtt{odd}}(F\ n) + \mathtt{p}_{\mathtt{even}}(F\ n)\big) \to \mathbf{2}$ be the function that sends `inl _` : $\mathtt{p}_{\mathtt{odd}}(F\ n) + \mathtt{p}_{\mathtt{even}}(F\ n)$ to 1 and `inr _` to 0. It's easy to see that the following holds

$$q : \prod_{n:\mathbb{N}} \big((e \circ f''\ n = 1) \to \mathtt{p}_{\mathtt{odd}}(F\ n)\big) \times \big((e \circ f''\ n = 0) \to \mathtt{p}_{\mathtt{even}}(F\ n)\big)$$

By $G.\mathtt{snd}$ we have that there exist $m, k : \mathbb{N}$ such that we have

$$l : F(m, 2k + e \circ f''\ m) = 1$$

This should normally be truncated, but given the context, we are allowed to drop it.

We then take cases on $(e \circ f''\ m = 0) + (e \circ f''\ m = 1)$.

- If $e \circ f''\ m = 0$ then by $q\ m$ we have $\mathtt{p}_{\mathtt{even}}(F\ m)$. This contradicts l.
- Similarly, $e \circ f''\ m = 1$ is also in contradiction with l.

In either case we reach falsum, concluding the proof. $\square$

Chapter 5

Extensional Type Theory

There exists a model of extensional type theory that validates the following 'Independence Principle'

$$\Big(\prod_{s:\mathbf{2}^{\mathbb{N}}} P\ s \to \Big(\prod_{n:\mathbb{N}} s\ n = 0\Big) \to \Big\| \sum_{z:\mathbb{N}} Q\ s\ z \Big\|\Big)$$

$$\to \Big(\prod_{s:\mathbf{2}^{\mathbb{N}}} P\ s \to \Big\| \sum_{z:\mathbb{N}} \Big(\prod_{n:\mathbb{N}} s\ n = 0\Big) \to Q\ s\ z \Big\|\Big)$$

where $P : \mathbf{2}^{\mathbb{N}} \to \texttt{Prop}$ and $Q : \mathbf{2}^{\mathbb{N}} \to \mathbb{N} \to \texttt{Prop}$ are families of propositions. Putting it plainly, if the left hand side of the above is true, then $z : \mathbb{N}$ does not depend on the proof of the constantness of s to 0.

The same model also validates Markov's Principle

$$\Big(\prod_{n:\mathbb{N}} (P\ n \vee \neg P\ n)\Big) \to \Big(\neg \prod_{n:\mathbb{N}} \neg P\ n\Big) \to \Big\| \sum_{n:\mathbb{N}} P\ n \Big\|$$

where $P : \mathbb{N} \to \texttt{Prop}$ is a family of propositions over $\mathbb{N}$. Informally, Markov's Principle states that if we have a collection of decidable propositions and the fact that not all of them are false, then one of them must be true.

In this section we work in the context of this model of Extensional Type Theory.

Let $A :\equiv \sum_{a:\mathbf{2}^{\mathbb{N}}} \texttt{atMost1one}\ a$. When referring to elements of A we implicitly mean the first part of the pair. Let

$$B : A \to \mathcal{U}$$

$$B :\equiv \lambda\ a.\ \mathtt{p}_{\mathtt{odd}}\ a + \mathtt{p}_{\mathtt{even}}\ a$$

We would like to have access to the following instance of IP in our intensional model as well.

$$\prod_{a:A} \prod_{f:B\ a \to \mathbb{N}} \Bigg(\Bigg(\prod_{s:\mathbf{2}^{\mathbb{N}}} \Big(\big(\prod_{n:\mathbb{N}} s(n) = a(2 \cdot n) \big) \vee \big(\prod_{n:\mathbb{N}} s(n) = a(2 \cdot n + 1) \big) \Big) \to \Big(\prod_{n:\mathbb{N}} s\ n = 0 \Big) \to \Big\| \sum_{z:\mathbb{N}} \prod_{p:B\ a} f^*\ p = z \Big\| \Bigg)$$

$$\to \Bigg(\prod_{s:\mathbf{2}^{\mathbb{N}}} \Big(\big(\prod_{n:\mathbb{N}} s(n) = a(2 \cdot n) \big) \vee \big(\prod_{n:\mathbb{N}} s(n) = a(2 \cdot n + 1) \big) \Big) \to \Big\| \sum_{z:\mathbb{N}} \Big(\prod_{n:\mathbb{N}} s\ n = 0 \Big) \to \prod_{p:B\ a} f^*\ p = z \Big\| \Bigg) \Bigg)$$

The fact that $\mathbb{N}$ is modal, follows from this in Intensional Type Theory. Our plan of action shall be to find a consequent of it, IP', strong enough for our purposes, that has the right form so that by Theorem 6.1 of [1] we have a model of intensional type theory that satisfies IP'.

We drop the truncation around $\|\sum_{z:\mathbb{N}} \prod_{p:B\ a} f^*\ p = z\|$ and since this is the consequent of the antecedent, the resulting statement is weaker than the original IP. We then uncurry 4 times to reach what we propose as our IP', a function that takes four arguments in the form of a quaternary dependent pair

$$\begin{aligned}
a :&\quad A\\
f :&\quad B\ a \to \mathbb{N}\\
\_ :&\quad \Big(\prod_{\bar{s}:\mathbf{2}^{\mathbb{N}}}\Big(\big(\prod_{n:\mathbb{N}}\bar{s}(n) = a(2\cdot n)\big) \vee \big(\prod_{n:\mathbb{N}}\bar{s}(n) = a(2\cdot n+1)\big)\Big) \to\\
&\qquad \Big(\prod_{n:\mathbb{N}}\bar{s}\ n = 0\Big) \to \sum_{z:\mathbb{N}}\prod_{p:B\ a} f^*\ p = z\Big)\\
r :&\quad \sum_{s:\mathbf{2}^{\mathbb{N}}}\Big(\big(\prod_{n:\mathbb{N}}s(n) = a(2\cdot n)\big) \vee \big(\prod_{n:\mathbb{N}}s(n) = a(2\cdot n+1)\big)
\end{aligned}$$

and has return type

$$\Big\|\sum_{z:\mathbb{N}}\prod_{p:B\ a} f^*\ p = z\Big\|$$

Lemma 5.0.1. $\mathbb{N}$ *is* $\|B\|$*-null.*

Proof. Given $a : A$ and $f : \|B\|\ a \to \mathbb{N}$ we need to prove that there exists unique $f' : \mathbf{1} \to \mathbb{N}$ through which f factors, in the sense that $f = f' \circ g$, where $g : C\ a \to \mathbf{1}$ is the sole inhabitant of its function space. Functions with $\mathbf{1}$ as their domain are constant. We therefore need to find some $z : \mathbb{N}$ so that $\lambda\ \_.\ z$ is f'. To that end, we will use the following instance of the Independence Principle

$$\Big(\prod_{s:\mathbf{2}^{\mathbb{N}}}\Big(\big(\prod_{n:\mathbb{N}}s(n) = a(2\cdot n)\big) \vee \big(\prod_{n:\mathbb{N}}s(n) = a(2\cdot n+1)\big)\Big) \to \Big(\prod_{n:\mathbb{N}}s\ n = 0\Big) \to \Big\|\sum_{z:\mathbb{N}}\prod_{p:B\ a} f^*\ p = z\Big\|\Big) \tag{5.1}$$

$$\to \Big(\prod_{s:\mathbf{2}^{\mathbb{N}}}\Big(\big(\prod_{n:\mathbb{N}}s(n) = a(2\cdot n)\big) \vee \big(\prod_{n:\mathbb{N}}s(n) = a(2\cdot n+1)\big)\Big) \to \Big\|\sum_{z:\mathbb{N}}\Big(\prod_{n:\mathbb{N}}s\ n = 0\Big) \to \prod_{p:B\ a} f^*\ p = z\Big\|\Big) \tag{5.2}$$

where f^* is the composition $f \circ C^B$, with C^B being the canonical function $B\ a \to C\ a$. Note that the left hand side 5.1 is provable. So we have a witness of the right hand side 5.2. By providing the odd s_1 and even s_2 subsequences of a to it, we get hold of $z_1 : \mathbb{N}$ and $z_2 : \mathbb{N}$ respectively. Equality on $\mathbb{N}$ is decidable, therefore $(z_1 = z_2) + (z_1 \neq z_2)$ is provable.

We will first prove that the existence of a candidate for f' follows from both constituents of the coproduct.

- First the case where $z_1 = z_2$. We arbitrarily pick z_1 and propose $f' :\equiv \lambda\ \_.\ z_1 : \mathbf{1} \to \mathbb{N}$. By function extensionality we reduce proving $f = f' \circ g$ to proving $f\ c = f' \circ g\ c$ for arbitrary $c : C\ a$.
  Since our goal is a mere proposition, we can act as if we have access to an actual $b : B\ a$. We then take cases on b.
  - We first consider the case $\mathtt{p_{odd}}\ a$. Through this we can still get access to some actual $b' : B\ a$. The odd subsequence is constantly 0, which we can provide to the rhs of the IP alongside with b' (and s_1 as the first argument) to deduce $f^*\ b' = z_1$. Using this we then prove $f\ c = z_1$. We concatenate this with $f' \circ g\ c = z_1$ to conclude this case.
  - The $\mathtt{p_{even}}$ case is very similar. The only difference is that we provide s_2 as the first argument to the rhs of the IP and then have $z_1 = z_2$ be part of the concatenation of paths when we prove homotopy between f and $f' \circ g$.
- Now consider the case $z_1 \neq z_2$. For any $n : \mathbb{N}$, $a\ n = 1$ is decidable. Suppose that $\prod_{n:\mathbb{N}} a\ n \neq 1$. Then a and by extension s_1 and s_2 are constantly 0. By the rhs of the IP we then have $z_1 = f\ b = z_2$ where $b : B\ a$ is clearly available to us at this point. This contradicts $z_1 \neq z_2$. We have proven $\neg \prod_{n:\mathbb{N}} a\ n \neq 1$. By Markov's Principle there exists $n_1 : \mathbb{N}$ such that $a\ n_1 = 1$. We pick the subsequence s_0 with parity opposite to n_1 and let f' be always equal to the corresponding z_0. We prove $\prod_{n:\mathbb{N}} s_0\ n = 0$ by induction on $\mathbb{N}$ and cases on $s_0\ n$. In the case where $s_0\ n = 1$ we can reach falsum because n will have to be both even and odd (in a) by `atMost1one` a. Ex falso trivializes this case. Now that we have established this too, we have enough arguments for the rhs of IP to output the desired equality $f\ b = z_0$ which proves homotopy between f and $(\lambda\ \_.\ z_0) \circ g$.

We still need to prove the uniqueness of f'. Since $\mathbf{1}$ is finite and equality on $\mathbb{N}$ is decidable, we can decide equality on the function space

$$p : \prod_{f_1, f_2 : \mathbf{1} \to \mathbb{N}} f_1 = f_2 + f_1 \neq f_2$$

We will use this to prove that $\sum_{h:\mathbf{1}\to\mathbb{N}} f = h \circ g$ is a mere proposition, which entails the uniqueness of f'. Consider $h_1, h_2 : \mathbf{1} \to \mathbb{N}$ such that $f = h_i \circ g,\ i \in \{1, 2\}$. By p we have $(h_1 = h_2) + (h_1 \neq h_2)$. We use induction on the coproduct. The left case is trivial. For the right case we have $q : h_1 \neq h_2$ and we will first try to prove $B\ a \to \bot$ as a stepping stone. Consider $b : B\ a$. Since $h_1 \circ g = f = h_2 \circ g$, by homotopy $h_1(g\ b) = h_2(g\ b)$. We can then prove $\prod_{*:\mathbf{1}} h_1(*) = h_2(*)$. By function extensionality $h_1 = h_2$. But this contradicts q, so we reach $\bot$. We've just proven $r : B\ a \to \bot$. We still have to prove $h_1 = h_2$, the main goal. By Lemma 4.0.3 we have $(B\ a \to \bot) \to \bot$ and by r we get $\bot$. We conclude the proof by Ex Falso. □

Lemma 5.0.2. *If C, D are $\|B\|$-null then so is $C + D$.*

Proof. We need to show that for $f : \|B\ a\| \to C + D$ there exists unique $f' : \mathbf{1} \to C + D$ such that $f = f' \circ e$, where $e : \|B\ a\| \to \mathbf{1}$ is the sole inhabitant of its function space. We define

$$g : C + D \to \mathbf{2}$$

using components

$$g_C :\equiv \lambda\ \_.\ 0 : C \to \mathbf{2}$$

$$g_D :\equiv \lambda\ \_.\ 1 : D \to \mathbf{2}$$

Same as we did for Lemma 5.0.1 we can lift a variant of IP, that will help us in our endeavor, from the model of Extensional Type Theory. This variant takes the following arguments in the form of a 4-product

$$\begin{aligned}
a &: && A \\
\bar{f} &: && B\ a \to C + D \\
\_ &: && \Big(\prod_{\bar{s}:\mathbf{2}^{\mathbb{N}}} \Big(\big(\prod_{n:\mathbb{N}} \bar{s}(n) = a(2 \cdot n)\big) \vee \big(\prod_{n:\mathbb{N}} \bar{s}(n) = a(2 \cdot n + 1)\big)\Big) \to \\
& && \Big(\prod_{n:\mathbb{N}} \bar{s}\ n = 0\Big) \to \sum_{z:\mathbf{2}} \prod_{b:B\ a} g(\bar{f}\ b) = z\Big) \\
r &: && \sum_{s:\mathbf{2}^{\mathbb{N}}} \Big(\big(\prod_{n:\mathbb{N}} s(n) = a(2 \cdot n)\big) \vee \big(\prod_{n:\mathbb{N}} s(n) = a(2 \cdot n + 1)\big)
\end{aligned}$$

and returns

$$\Big\| \sum_{z:\mathbf{2}} \prod_{b:B\ a} g(\bar{f}\ b) = z \Big\|$$

We provide this with the arguments required, where $\bar{f}$ is f composed with the truncation map $|*| : B\ a \to \|B\ a\|$, to get a witness

$$\bar{z} : \sum_{z:\mathbf{2}} \prod_{b:B\ a} g(\bar{f}\ b) = z$$

Note that we dropped the truncation since our goal is a mere proposition, therefore we can act as if we have an actual $\bar{z}$. Using $\bar{z}$ we can prove that

$$y : \sum_{z:\mathbf{2}} \prod_{b:\|B\ a\|} g(f\ b) = z$$

We can prove that

$$y = 0 + y = 1$$

We use induction on this coproduct to prove our goal. Without loss of generality, we argue only for the case of $y = 0$. As a first step towards providing a candidate for f', we will try to construct some $h : \|B\ a\| \to C$ such that $\texttt{inl} \circ h = f$.

$$\begin{tikzcd}
& \mathbf{1} \arrow[d, dashed, "h'"'] \arrow[ddr, dashed, bend left, "f'"] & & \\
& C \arrow[dr, "\texttt{inl}"] & & \\
B\ a \arrow[r, "|*|"] \arrow[rr, bend right, "\bar{f}"'] & \|B\ a\| \arrow[r, "f"] \arrow[u, dashed, "h"] \arrow[uu, bend left, "e"] & C + D \arrow[r, "g"] & \mathbf{2}
\end{tikzcd}$$

Let $b : \|B\ a\|$. First, note that we can prove

$$\zeta : \prod_{a:C+D} \Big(\big(\sum_{c:C} \texttt{inl}\ c = a \big) + \big(\sum_{d:D} \texttt{inr}\ d = a \big) \Big)$$

by induction on $C + D$. We then take cases on

$$\zeta(f\ b) : \big(\sum_{c:C} \texttt{inl}\ c = f\ b \big) + \big(\sum_{d:D} \texttt{inr}\ d = f\ b \big)$$

- If $\bar{c} : \sum_{c:C} \texttt{inl}\ c = f\ b$, we define $h\ b :\equiv \bar{c}$. Clearly $\texttt{inl} \circ h = f$.
- If $\sum_{d:D} \texttt{inl}\ d = f\ b$, then $g(f\ b) = 1$ but since $y = 0$ we also have $g(f\ b) = 0$. Contradiction. This case is resolved by Ex Falso.

Since C is $\|B\|$-null, there exists $h' : \mathbf{1} \to C$ such that $h = h' \circ e$. Observe that $\texttt{inl} \circ h' : \mathbf{1} \to C + D$, furthermore $(\texttt{inl} \circ h') \circ e = f$. We have found a valid candidate for f', what is left is to show it's unique.

Consider any f' candidate. We can construct a $f'_C : \mathbf{1} \to C$ so that f' factors through it, $f' = \texttt{inl} \circ f'_C$, with our construction being similar to that of h earlier. We want to show that $f' = \texttt{inl} \circ h'$ or $\texttt{inl} \circ f'_C = \texttt{inl} \circ h'$. We will show that $h' = f'_C$. Since C is $\|B\|$-null, $\sum_{k:\mathbf{1} \to C} k \circ e = h$ is a mere proposition. Therefore showing $h' = f'_C$ is reduced to showing $f'_C \circ e = h$. We use function extensionality. We know that for arbitrary $b : \|B\ a\|$, $\texttt{inl}(f'_C \circ e\ b) = \texttt{inl}(h\ b)$. By 2.12.1 of [2] we have $(\texttt{inl}\ a_1 = \texttt{inl}\ a_2) \simeq (a_1 = a_2)$. We can then deduce that $f'_C \circ e\ b = h\ b$. This concludes the proof of uniqueness of h'.

□

Bibliography

[1] Andrew Swan and Taichi Uemura. *On Church's Thesis in Cubical Assemblies*. 2019. eprint: `arXiv:1905.03014`.

[2] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study: `https://homotopytypetheory.org/book`, 2013.