

Univalence Axiom via Cubical Type Theory

Yiming Xu

09/11/2018

0 Introduction

Univalence axiom is one of the most significant theorem of homotopy type theory, it says that “being isomorphic is equivalent to being equal”. It is a very powerful theorem that helps a lot in formalization of mathematics in type theory. For instance, apply it directly allows us to identify homotopic equivalent types, and together with function extensionality it enables us to identify homotopic functions.

In this paper, we will give a model of the univalence axiom, that is, instead of have it as an axiom, we construct a model where the statement holds and prove it. This is achieved by introduce a way of doing type theory in “cubes”, and here is what we mean by “cubical type theory”.

This artical consists of three parts: Firstly, we give some basic definitions of cubical type theory, these definitions are all based on the dependent type theory. And then in the second section, we see what we can do with the things we defined in the first section by define some derived operation. And in the final section, we prove the statement of univalence axiom as a theorem with the construction we give in section one and two.

1 Basic Construction

The cubical type theory is based on standard type theory, which has:

$\Gamma, \Delta ::= () \mid \Gamma, x : A$

as rule of forming contexts. That is, the empty context is valid, and if $\Gamma \vdash A$, which means that A is a type under context Γ , then $\Gamma, x : A$ is a valid context.

The terms for \prod -types and \sum -types are:

$$t, u, A, B ::= x \mid \lambda x. t \mid t \ u \mid \prod_{x:A} B(x)$$

for \prod -types, and

$$\cdots \mid (t, u) \mid t_1 \mid t_2 \mid \sum_{x:A} B(x)$$

for \sum -types. Where t_1, t_2 denotes the projection on the first or second component, respectively.

Substitutions are written as $\sigma = (x_1/u_1, \dots, x_n/u_n)$, and $A\sigma$ means replacing any occurrence of x_i by u_i , renaming bounded variables if necessary.

To do cubical type theory we will introduce some new types as below.

1(a) The Free De Morgan Algebra \mathbb{I}

Assuming we are give a discrete infinite set of names representing directions, with elements i, j, k, \dots . Define \mathbb{I} to be the free De Morgan algebra over the set. Elements in \mathbb{I} are of the form:

$$r, s ::= 0 \mid 1 \mid i \mid 1 - r \mid r \wedge s \mid r \vee s$$

.

and satisfy the rules:

$$1 - 0 = 1, 1 - 1 = 0, 1 - (r \vee s) = (1 - r) \wedge (1 - s), 1 - (r \wedge s) = (1 - r) \vee (1 - s)$$

\mathbb{I} is a bounded distributive lattice with top element 1 and bottom element 0, elements in \mathbb{I} can be viewed as elements in the interval $[0, 1]$, and the \wedge, \vee can be viewed as max and min, respectively, so actually it is not a general fact that $(1 - i) \wedge i = 0, (1 - i) \vee i = 1$.

Now we add a new rule regarding extension of a context:

$$\frac{\Gamma \vdash, i \notin \text{dom}(\Gamma)}{\Gamma, i : \mathbb{I} \vdash}$$

Which says if Γ is a valid context, then Γ, i with $i : \mathbb{I}$ where $i : \mathbb{I}$ is a valid context.

When writting $\Gamma \vdash r : \mathbb{I}$, it means Γ is a valid context and $r : \mathbb{I}$ depends only on the names declared in Γ .

1(b) Path Type

Bases on dependent type theory as described earlier, we add the terms of path type as below:

$$t, u, A, B ::= \dots \mid \text{Path } A \ t \ u \mid \langle i \rangle t \mid t \ r$$

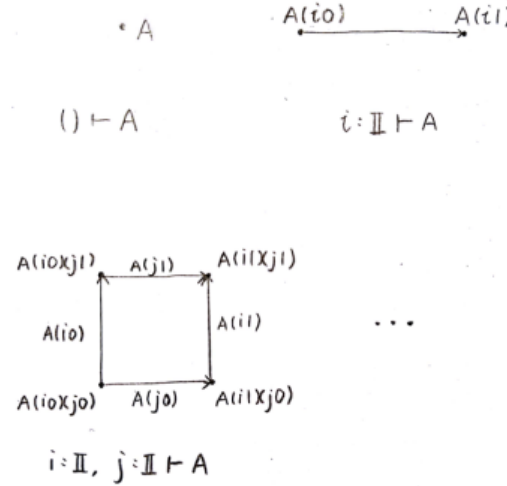
For a type A with $t, u : A$, $\text{Path } A \ t \ u$ is the type of paths in A from t to u . For $p : \text{Path } A \ t \ u$, $\langle i \rangle p$ means regarding the path as a function with variable i . And $p \ r$ means apply this function at point p .

The inference rules of the path type are as follows, note that $(i0), (i1)$ are just shorthands of the substitutions $(i/0), (i/1)$, we will use this notation through all of this article.

$$\begin{array}{c} \frac{\Gamma \vdash A \quad \Gamma \vdash t : A \quad \Gamma \vdash u : A}{\Gamma \vdash \text{Path } A \ t \ u} \\[10pt] \frac{\Gamma \vdash t : \text{Path } A \ u_0 \ u_1 \quad \Gamma \vdash i : \mathbb{I}}{\Gamma \vdash t \ r : A} \end{array} \qquad \begin{array}{c} \frac{\Gamma \vdash A \quad \Gamma, i : \mathbb{I} \vdash t : A}{\Gamma \vdash \langle i \rangle t : \text{Path } A \ t(i0) \ t(i1)} \\[10pt] \frac{\Gamma \vdash A \quad \Gamma, i : \mathbb{I} \vdash t : A}{\Gamma \vdash \langle i \rangle t : \text{Path } A \ t(i0) \ t(i1)} \end{array}$$

$$\frac{\Gamma, i : \mathbb{I} \vdash t \ i = u \ i : A}{\Gamma \vdash t = u : \text{Path } A u_0 u_1} \quad \frac{\Gamma \vdash t : \text{Path } A u_0 u_1}{\Gamma \vdash t \ 0 = u_0 : A} \quad \frac{\Gamma \vdash t : \text{Path } A u_0 u_1}{\Gamma \vdash t \ 1 = u_1 : A}$$

By “a type depends on a name”, we mean that from the description of the type A , there exists a variable that determines how A is defined. For instance, if A is defined to be the type of vectors in \mathbb{N} with n components, then A depends on the name n . A type depends on n -names can be viewed as an n -cube, as shown in the digrams below.



Note that just as we have constant functions, when we have a type A which is no depend on any name, we can certainly regard it as a degenerate type depending on the name i . in this case, $A(i) = A$ for any $i : \mathbb{I}$. We will have $\Gamma, i : \mathbb{I} \vdash A$ very frequently in this artical. In this case, we can regard the type A and being parametrized over the interval $[0, 1]$ and think such an A as $\sum_{i:\mathbb{I}} A(i)$, hence $A(i0)$ and $A(i1)$ denotes the fibre over 0 and 1, respectively. The substitution (i/j) is just renaming the variable of the path regarded as a function, and $(i/1 - i)$ corresponds to reversing the direction of a path.

1(c) Face Lattice

Given a set of names i, j, k, \dots , the elements of the face lattice on the set are of the form:

$$\varphi, \psi ::= 0_{\mathbb{F}} \mid 1_{\mathbb{F}} \mid (i = 0) \mid (i = 1) \mid \varphi \wedge \psi \mid \varphi \vee \psi$$

It is the distributive lattice generated by elements of the form $i = 0$ and $i = 1$, and the relation $(i = 0) \wedge (i = 1) = 0_{\mathbb{F}}$. For any face formula, it is equal to a disjunction of face formulas.

The elements of the face lattice are called face formulas, geometrically, any face formula is a subpolyhedra of a cube. In particular, $0_{\mathbb{F}}$ is the empty subset of the cube, and $1_{\mathbb{F}}$ is the whole cube.

We now introduce a restriction operation on context:

$$\frac{\Gamma \vdash \varphi : \mathbb{F}}{\Gamma, \varphi \vdash}$$

It says that if φ is a face formula which depends only on the names declared in Γ , then restriction on some subpolyhydra is also a valid context. If $\Gamma = i : \mathbb{I}, j : \mathbb{I}$, then Γ can be thought as a square. And if $\Gamma \vdash A$, then the displayed below are examples of $\Gamma, \varphi \vdash A$ for some face formula φ .

$$\begin{array}{l}
 i : \mathbb{I}, (i=0) \vee (i=1) \vdash A \\
 i : \mathbb{I}, j : \mathbb{I}, (i=0) \vee (j=1) \vdash A \\
 i : \mathbb{I}, j : \mathbb{I}, (i=0) \vee (i=1) \wedge (j=0) \vdash A
 \end{array}$$

2 Derived Operations

2(a) Composition

If $\Gamma, \varphi \vdash u : A$, then we write $\Gamma \vdash a : A[\varphi \mapsto u]$ as an abbreviation for $\Gamma \vdash a : A$ and $\Gamma, \varphi \vdash a = u : A$. More generally, when φ 's are face formulas, $\Gamma \vdash a : A[\varphi_1 \mapsto u_1, \dots, \varphi_n \mapsto u_n]$ denotes $\Gamma \vdash a : A$ and $\Gamma, \varphi_l \vdash a = u_l : A$ for any $l = 1, \dots, n$.

For example, when $\varphi = (i = 0) \vee (i = 1)$, $\Gamma, \varphi \vdash u : A$, then u consists two pieces of informations- u_0 at $i = 0$ and u_1 at $i = 1$. An element $a : A[\varphi \mapsto u]$ encodes the information of a path from u_0 to u_1 .

We have an operation:

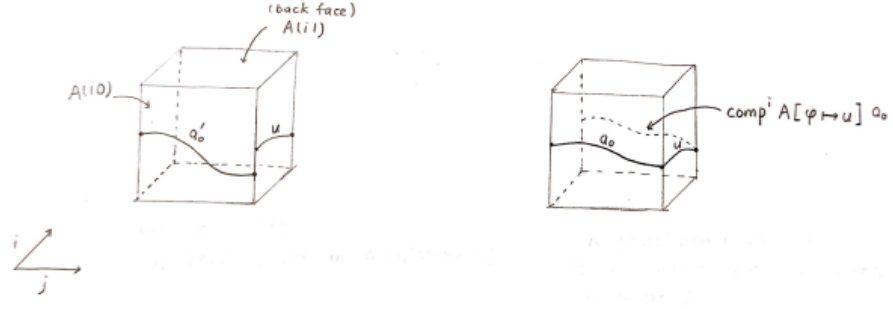
$$\frac{\Gamma \vdash \varphi : \mathbb{F} \quad \Gamma, i : \mathbb{I} \vdash A \quad \Gamma, \varphi, i : \mathbb{I} \vdash u : A \quad \Gamma \vdash a_0 : A(i0)[\varphi \mapsto u(i0)]}{\Gamma \vdash \text{comp}^i A[\varphi \mapsto u] a_0 : A(i1)[\varphi \mapsto u(i1)]}$$

The operation above gives the principle that if we have a partial path u with i bounded on the polyhydra φ , and there exists an element $a_0 : A(i0)$ that restrict to $i = 0$ agrees with the endpoint at $i = 0$ of u , that is, a_0 sticks on the tail of u and extends it. Then we can construct an element on $A(i1)$ that extends u at another endpoint. Take an example, for $\varphi = (j = 1)$. a'_0 below gives an example that is in $A(i0)$ but not in $A(i0)[\varphi \mapsto u(i0)]$, in this case we can do nothing. But a_0 satisfies the condition of being in $A(i0)[\varphi \mapsto u(i0)]$, it serves an evidence that $u(i0)$ extends off the context φ , and hence by the definition of composition operation we obtain an element on $A(i1)$ which is an evidence that $u(i1)$ extends off the context φ , the picture below gives a visualization of this example.

For a remark, note that when $\varphi = 0_{\mathbb{F}}$, there is no "agree at endpoint" condition to meet, so we obtain an element $\text{comp}^i A[0_{\mathbb{F}} \mapsto []] a : A(i1)$ for free, this is just the transportation operation along the path from 0 to 1.

2(b) Kan Filling

Once $\text{comp}^i A[\varphi \mapsto u] a_0$ is defined, we have an operation defined as:

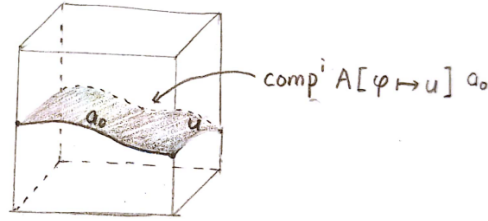


$$\Gamma, i : \mathbb{I} \vdash \text{fill}^i A[\varphi \mapsto u] a_0 = \text{comp}^j A(i/i \wedge j)[\varphi \mapsto u(i/i \wedge j)], (i=0) \mapsto a_0 : A$$

Where $j \notin \text{dom}(\Gamma)$. Let $\Gamma, i : \mathbb{I} \vdash v = \text{fill}^i A[\varphi \mapsto u] a_0 : A$. Then by definition of composition we have $v : A(i/i \wedge 1)[\varphi \mapsto u(i/i \wedge 1), i=0 \mapsto a_0]$. Note $i/i \wedge 1 = i/i$ so the substitution does nothing. It follows directly that $v(i0) = a_0$ and $\Gamma, \varphi, i : \mathbb{I} \vdash v = u : A$. Also we can check:

$$v(i1) = \text{comp}^j A(i/j)[\varphi \mapsto u(i/j)] a_0 = \text{comp}^i A[\varphi \mapsto u] a_0$$

So $\langle i \rangle v$ gives a path from a_0 to $\text{comp}^i A[\varphi \mapsto u] a_0$, as shown in the shaded region below:



2(c) contr from Contractible Types

Just as in the HoTT book, let $\text{isContr} := \sum_{x:A} \prod_{y:A} \text{Path } A \ x \ y$, so having an element in $\text{isContr } A$ means the type A is contractible.

We define the operation contr by:

$$\frac{\Gamma \vdash p : \text{isContr } A \quad \Gamma, \varphi \vdash u : A}{\text{contr } p[\varphi \mapsto u] = \text{comp}^i A[\varphi \mapsto p_2 \ u \ i] p_1 : A[\varphi \mapsto u]}$$

Lemma 2.1 If we have an operation

$$\frac{\Gamma \vdash A \quad \Gamma, \varphi \vdash u : A}{\text{contr}[\varphi \mapsto u] : A[\varphi \mapsto u]}$$

Then we can construct an element of $\text{isContr } A$.

Proof. It suffices to pick an element x in A such that for any element y in A there exists a path connecting x and y , we pick $x := \text{contr}[]$, so we know nothing about x except for the trivial fact that $x : A$. Take any $y : A$, to find the path connecting x and y , we take a fresh name $i : \mathbb{I}$, take $\varphi := (i = 0) \vee (i = 1)$, $u = [i = 0 \mapsto x, i = 1 \mapsto y]$, so the operation gives us an element $\text{contr}[\varphi \mapsto u] : A[\varphi \mapsto u]$, which is as discussed before, a path connecting x and y . \square

corollary 2.2 Suppose for any partial element $u : A$ defined on an extent φ , we can find a path-equal partial element u' and a total element that restricts to u' , then $\text{isContr } A$ is inhabited.

Proof. It is enough to define an operation as in the theorem above. Given $u : A$, the assumption that there exists a total element v which is restrict to φ is path equal to u translates to $\Gamma \vdash v : A$ and $\Gamma, \varphi \vdash p : \text{Path } A \ u \ v$, note that we have $\Gamma, \varphi, i \vdash p(i/1 - i) : A$ and $\Gamma \vdash v : A[\varphi \mapsto p \ 1 (= v)]$, we can define $u' := \text{comp}^i A[\varphi \mapsto p(i/1 - i)] \ v : A[\varphi \mapsto p \ 0 (= u)]$, so we obtain a desired operation. \square

2(d) equiv from Equivalences

We define $\text{isEquiv } T \ A \ f = \prod_{y:A} \text{isContr}(\sum_{x:T} \text{Path } A \ y \ (f \ x))$. Note that for each $y : A$, $\sum_{x:T} \text{Path } A \ y \ (f \ x)$ is the type of fibre over y , so an inhabitant in $\text{isEquiv } T \ A \ f$ means that any fibre of f is contractible. And we define $\text{Equiv } T \ A := \sum_{f:T \rightarrow A} \text{isEquiv } T \ A \ f$. We may write $f : \text{Equiv } T \ A$ to express the fact that f is an equivalence from T to A , and write $f \ t$ instead of $f_1 \ t$.

If $\Gamma \vdash f : \text{Equiv } T \ A$, we define an operation

$$\frac{\Gamma, \varphi \vdash t : T \quad \Gamma \vdash a : A \quad \Gamma, \varphi \vdash p : \text{Path } A \ a \ (f \ x)}{\Gamma \vdash \text{equiv } f \ [\varphi \mapsto (t, p)] \ a := \text{contr}(f_2 \ a)[\varphi \mapsto (t, p)] : (\sum_{x:T} \text{Path } A \ a \ (f \ x))[\varphi \mapsto (t, p)]}$$

Lemma 2.3 If we have an operation:

$$\frac{\Gamma \vdash f : T \rightarrow A \quad \Gamma, \varphi \vdash t : T \quad \Gamma \vdash a : A \quad \Gamma, \varphi \vdash p : \text{Path } A \ a \ (f \ x)}{\Gamma \vdash \text{equiv } f \ [\varphi \mapsto (t, p)] \ a := \text{contr}(f_2 \ a)[\varphi \mapsto (t, p)] : (\sum_{x:T} \text{Path } A \ a \ (f \ x))[\varphi \mapsto (t, p)]}$$

then f is an equivalence.

Proof. Proving f is an equivalence amounts to construct an element of $\text{isEquiv } T \ A \ f$, which again amounts to construct an element of $\text{isContr}(\sum_{x:T} \text{Path } A \ y \ (f \ x))$ for each $y : A$. The result then follows directly from **Lemma 2.1**. \square

2(e) Gluing

This section introduces a bunch of types regarding gluing, together with some intuitions.

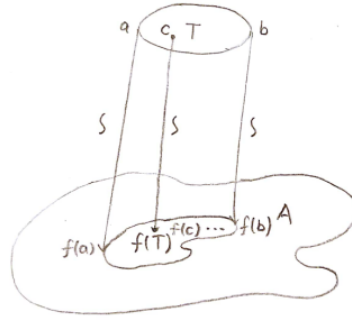
2(e).1 Glue

Given $\Gamma \vdash A$, $\Gamma, \varphi \vdash T$ and $\Gamma, \varphi \vdash f : \text{Equiv } T \rightarrow A$ (To be clearer, this means $f : T \rightarrow A$ is defined on the type on the extent φ and is an equivalence from T to the image of f , it does not say that T is equivalent to A .), we have a type $\Gamma \vdash \text{Glue}[\varphi \mapsto (T, f)] A$. In particular, when $\varphi = 1_{\mathbb{F}}$, which means $\Gamma \vdash f : \text{Equiv } T \rightarrow A$, we have $\Gamma \vdash \text{Glue}[1_{\mathbb{F}} \mapsto (T, f)] A = T$. Formally, the two rules are expressed as:

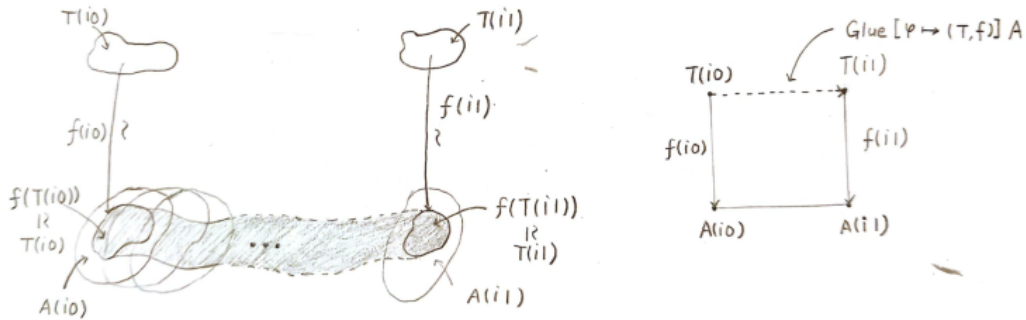
$$\frac{\Gamma \vdash A \quad \Gamma, \varphi \vdash T \quad \Gamma, \varphi \vdash f : \text{Equiv } T \rightarrow A}{\Gamma \vdash \text{Glue}[\varphi \mapsto (T, f)] A}$$

$$\frac{\Gamma \vdash T \quad \Gamma, \Gamma \vdash f : \text{Equiv } T \rightarrow A}{\Gamma \vdash \text{Glue}[1_{\mathbb{F}} \mapsto (T, f)] A = T}$$

The Glue type is indeed a special case of the “gluing construction” of spaces as in algebraic topology: Think about two spaces T and A , and a map $f : T \rightarrow A$, by the fact that f is a partial equivalence, we can identify $f(T) \subseteq A$ with T , and “glue” T with $f(T)$ along the equivalence f . Explicitly, the Glue space formed in this way is $T \amalg A/t \sim f(t)$, which is drawn as the picture below:



The notation $\text{Glue}[\varphi \mapsto (T, f)]$ is reasonable: Restricted to the extent where T lives in, the glued type is just T itself, the gluing just constructs a bridge using the equivalence to link the T to the A , and hence extend the type T . For instance, when $\varphi = (i = 0) \vee (i = 1)$, the dashed line together with its two end points T_0, T_1 in the diagram on the right hand side below:



is the type $\text{Glue}[i = 0 \mapsto (T_0, f(i0)), i = 1 \mapsto (T_1, f(i1))]$, which corresponds to the shaded region of the graph on the left hand side. As we can see, when $i = 0$ or $i = 1$ it is just the type T . Actually, with

the commutative digram above, note that equivalence is “reversible”, we can think the dotted line as the composition of the three lines $T_0 \rightarrow A(i0)$, $A(i0) \rightarrow A(i1)$ and $A(i1) \rightarrow T_1$, where the last one is obtained by reversing equivalence. The key observation that how does Glue yields a total type is that it takes elements from the total type A to fill the “missed part” of the partial type T , and we make this doable by embedding T into A via the equivalence f .

2(e).2 unglue

Given $\Gamma \vdash A$, $\Gamma, \varphi \vdash T$ and $\Gamma, \varphi \vdash f : \text{Equiv } T \ A$, as discussed above we can extend the type T to be a total type $\text{Glue}[\varphi \mapsto (T, f)]$, and also we can extend the function f to a total function as well. The extended function unglue is a function $\text{Glue}[\varphi \mapsto (T, f)] \ A \rightarrow A$.

As we expect, we have the rule:

$$\frac{\Gamma \vdash b : \text{Glue}[\varphi \mapsto (T, t)] \ A}{\Gamma \vdash \text{unglue } b : A[\varphi \mapsto f \ b]}$$

Which says that the function unglue restricted to φ where T is defined is just the function f itself.

2(e).3 glue

As mentioned before, given $\Gamma \vdash A$, $\Gamma, \varphi \vdash T$ and $\Gamma, \varphi \vdash f : \text{Equiv } T \ A$, to form the Glue type, we pick elements in A to fill the “missed part” of T , that is, we glue elements in A onto the type T to extend it. We have a rule that describes this operation:

$$\frac{\Gamma, \varphi \vdash f : \text{Equiv } T \ A \quad \Gamma, \varphi \vdash t : T \quad \Gamma \vdash a : A[\varphi \mapsto f \ t]}{\Gamma \vdash \text{glue}[\varphi \mapsto t] \ a : \text{Glue}[\varphi \mapsto (T, f)] \ A}$$

This rule says that if an element $a : A$ restricted to φ is equal to some $f \ t$, then a can be taken as the element that glued to the element $t : T$ and extend it.

The inference rules of glue and how it interact with Glue and unglue are:

$$\frac{\Gamma \vdash t : T \quad \Gamma \vdash f : \text{Equiv } T \ A}{\Gamma \vdash \text{glue}[1_{\mathbb{F}} \mapsto t] \ (f \ t) = t : T}$$

$$\frac{\Gamma \vdash b : \text{Glue}[\varphi \mapsto (T, f)] \ A}{\Gamma \vdash b = \text{glue}[\varphi \mapsto b] \ (\text{unglue } b) : \text{Glue}[\varphi \mapsto (T, f)] \ A}$$

$$\frac{\Gamma, \varphi \vdash f : \text{Equiv } T \ A \quad \Gamma, \varphi \vdash t : T \quad \Gamma \vdash a : A[\varphi \mapsto (T, f)] \ A}{\Gamma \vdash \text{unglue}(\text{glue}[\varphi \mapsto t] \ a) = a : A}$$

The first rule says: If T is already a total type then the second rule in 2.(e).1 says $\text{Glue}[1_{\mathbb{F}} \mapsto (T, f)] \ A = T$, and the definition of Equiv says that the fibre over any element $a : A$ is inhabited, which means

any element of A is an $f\ t$ for some $t : T$. Then the map glue which takes elements in A to construct elements in $\text{Glue}[1_{\mathbb{F}} \mapsto (T, f)]\ A = T$ will take $f\ t : A$ back to $t : T$, which is intuitive because if T is already a total type, we do not need any extra information from A to fill the T into a total type, so it just does this tautological thing.

The second rule and third rule together says that glue and unglue are inverses: Gluing back an element of which comes from unglue gives the same element as before the unglue operation. And if for constructing the Glue type we glue a term a to t , then ungluing this element from the glued type gives a back.

3 Proof of the Univalence Axiom

Having the constructions on the previous section by hand, we are now in the position of proving the univalence axiom.

Theorem 3.1 Given $\Gamma \vdash A, \Gamma, \varphi \vdash T$, and $\Gamma, \varphi \vdash f : \text{Equiv}\ T\ A$. Denote $B = \text{Glue}\ [\varphi \mapsto (T, f)]\ A$, then the map $\text{unglue} : B \rightarrow A$ is an equivalence.

Proof. By **Lemma 2.3**, it suffices to construct, given the conditions $\Gamma, \psi \vdash b : B, \Gamma \vdash u : A$ and $\Gamma, \psi \vdash \alpha : \text{Path}\ A\ u\ (\text{unglue}\ b)$, an element of $(\sum_{x:B} \text{Path}\ A\ u\ (\text{unglue}\ b))[\psi \mapsto (b, \alpha)]$, which amounts to give a pair of elements $\tilde{b} : B[\psi \mapsto b]$ and $\tilde{\alpha} : \text{Path}\ A\ u\ (\text{unglue}\ \tilde{b})[\psi \mapsto \alpha]$.

To construct the element, we start by applying the equiv operation: Using the fact that $\Gamma, \varphi \vdash f : T \rightarrow A$ is an equivalence, we obtain a pair of elements $\Gamma, \varphi \vdash t : T[\psi \mapsto b]$ and $\Gamma, \varphi \vdash \beta : \text{Path}\ A\ u\ (f\ t)[\psi \mapsto \alpha]$.

View A as a degenerate type, then A does not really depend on i and we have $A(i0) = A(i1) = A$. In particular, $\Gamma \vdash u : A[\varphi \mapsto \beta\ 0 = u, \psi \mapsto \alpha\ 0 = u]$, so the inference rule of composition allows us to construct an element $\tilde{a} = \text{comp}^i\ A[\varphi \mapsto \beta\ i, \psi \mapsto \alpha\ i] : A[\varphi \mapsto f\ t, \psi \mapsto \text{unglue}\ b]\ u$. Define $\tilde{b} := \text{glue}[\varphi \mapsto t]\ \tilde{a}$ and $\tilde{\alpha} := \text{fill}^i\ A[\varphi \mapsto \beta\ i, \psi \mapsto \alpha\ i]\ u$. We can check that they are indeed the pair of elements we want: We have

$$\Gamma, \psi \vdash \tilde{b} \stackrel{\text{note the type of } \tilde{a}}{=} \text{glue}[\varphi \mapsto t](\text{unglue}\ b) \stackrel{\text{note the type of } t}{=} \text{glue}[\varphi \mapsto b](\text{unglue}\ b) \stackrel{\text{inference rule of glue}}{=} b$$

And also as we have seen in section 2.(b) where properties of fill are stated, we have $\Gamma, \psi \vdash \tilde{\alpha} = \alpha : A$, as desired.

□

corollary 3.2 For any type $A : \mathcal{U}$ the type $C = \sum_{X:\mathcal{U}} \text{Equiv}\ X\ A$ is contractible.

Proof. By **corollary 2.2**, it suffices to prove that for partial element $\Gamma, \varphi \vdash (T, f) : C$, it is path equal to the restriction on φ of some total element of C . By unfolding the definition of C , a partial element $\Gamma, \varphi \vdash (T, f) : C$ consists of three pieces of information: a partially defined type $\Gamma, \varphi \vdash T$, a partially defined function $\Gamma, \varphi \vdash f : T \rightarrow A$, and a proof in $\Gamma, \varphi \vdash \text{isEquiv}\ T\ A\ f$. For such an element to be path equivalent to a total element, what we need is a total type $\Gamma \vdash X$, a function $\Gamma \vdash g : X \rightarrow A$ and a proof in $\Gamma \vdash \text{isEquiv}\ X\ A\ g$ that together gives an element in C that restricted on φ is path equal to (T, f) . We claim we can take $X = \text{Glue}[\varphi \mapsto (T, f)]\ A, f = \text{unglue}$. $\Gamma \vdash (\text{Glue}[\varphi \mapsto (T, f)]\ A, \text{unglue}) : C$ by **Theorem 3.1**, and $\Gamma, \varphi \vdash \text{Glue}[\varphi \mapsto (T, f)]\ A, \Gamma, \varphi \vdash \text{unglue}$ are given as inference rules in section 2.(e). So we have restricting the proof that $(\text{Glue}[\varphi \mapsto (T, f)]\ A, \text{unglue})$ is an equivalence indeed gives

us an element in the partial type $\Gamma, \varphi \vdash \text{isEquiv } T \ A \ f$, but $\text{isEquiv } T \ A \ f$ is a mere proposition as discussed in the HoTT book, so any elements in this type have a path links them. The path equal condition follows directly from here. \square

Theorem 3.3 (Univalence Axiom) For any type $A, B \in \mathcal{U}$, we have the canonical function $\text{Path } \mathcal{U} \ A \ B \rightarrow \text{Equiv } A \ B$ is an equivalence.

Proof. Firstly we explain where does the canonical function come from: By **Lemma 2.10.1** in the HoTT book, we have a function $\text{Path } \mathcal{U} \ A \ B \rightarrow A \simeq B$, Comparing the definitions $A \simeq B := \sum_{f:A \rightarrow B} \text{isequiv } f$ and $\text{Equiv } A \ B = \sum_{f:A \rightarrow B} \text{isEquiv } A \ B \ f$, the only difference is that the “ \simeq ” take the takes the definition of equivalence in the textbook takes the existance of left and right inverses as the proof of being an equivalence, where as “Equiv” here takes the contractible fibre condition as the proof of being an equivalence, the two characterization is equivalent as proved in chapter 4 in the HoTT book. So the $A \simeq B$ and $\text{Equiv } A \ B$ are equivalent. Hence the composing this equivalence to the function constructed in **Lemma 2.10.1** gives a function $\text{Path } \mathcal{U} \ A \ B \rightarrow \text{Equiv } A \ B$.

Our goal is to prove this canonical function is an equivalence for any type A, B , that is the same as for any type B the function $t : \prod_{A:\mathcal{U}} \text{Path } \mathcal{U} \ A \ B \rightarrow \text{Equiv } A \ B$ that assigns each A to the corresponding function as described above, is a fibrewise equivalence. Then by **Theorem 4.7.7** in the HoTT book, it suffices to prove that the induced total function $\text{total}(t) : \sum_{A:\mathcal{U}} \text{Path } \mathcal{U} \ A \ B \rightarrow \sum_{A:\mathcal{U}} \text{Equiv } A \ B$ as defined as **Definition 4.7.5** is an equivalence.

By **Lemma 3.11.3**, $\sum_{A:\mathcal{U}} \text{Path } \mathcal{U} \ A \ B$ is contractible, so it suffices to show $\sum_{A:\mathcal{U}} \text{Equiv } A \ B$ is contractible as well, but we have already prove this as **Theorem 3.2**, so we are done. \square