UCL Department of Philosophy
Coursework Coversheet

This sheet needs to be included at the beginning of all electronic submissions via Moodle.

Module code: (beginning PHIL.…)	PHIL0146
Module Title:	MA Philosophy Dissertation
Dissertation Title:	Identity type under homotopy type theory with univalent axiom
Word Count:	~11,000
Candidate Number: In format (ABCD1)	NMNZ6

I am aware that paragraphs from my essay or the complete essay may be used in anonymised form in helping future students to improve their essay writing skills.
Opt-out: do not allow material to be used in teaching essay writing to future students ☐

Declaration:

By submitting this coursework, I affirm it is the product of my effort alone and meets all College and Department regulations regarding student conduct, especially those regarding plagiarism and reuse of previous work. I acknowledge that my work will be subject to checks for irregularities, such as through on-line plagiarism detection software, e.g., Turn-It-In.

UCL Plagiarism regulations:
http://www.ucl.ac.uk/current-students/guidelines/plagiarism

Important Notes:

Your name must not appear anywhere on the submission or file name.

Include your candidate number on this cover sheet and in the file name of your upload.

Your candidate number is found in Portico.
It is always comprised of four letters and a number (i.e. ABCD1).

Identity Type under Homotopy Type Theory with Univalent Axiom

Contents

1 Introduction

Homotopy type theory (abbreviated as HoTT) as its name suggest is a type theory with a homotopical point of view. As a branch of type theory, it shares most common features with other Martin-Löf intuionistic type theories; some of the distinguishing features of HoTT are its identity type, higher inductive types and the univalent axiom. HoTT is intended as an alternative foundation of mathematics on par with ZFC set theory. With univalent axiom and identity type, HoTT is known for its slogan that "isomorphic objects are identitcal". Awodey argues that HoTT should be the choice of mathematical structuralists.[1] But higher inductive types provides a nice syntactic approach to geometry and it would be a pity if HoTT and, in particular, higher inductive types are only available to those who hold a structuralist view. In this paper, I will develop an alternative neutral and instrumentalist reading so that mathematicians with philosophy of mathematics standpoint other than structuralism can utilise HoTT. A short overview of HoTT is given in Appendix A.

2 Basic feature

Like other type theories, any term[1] in the language of HoTT has a type and a term is always associated with its type though usually its type is obvious from context so that its type information is ignored for typographical reasons; while technically, all terms are associated with its types. For example $1 : \mathbb{N}$ is understood to mean the term 1 is of type $\mathbb{N}$. When $\mathbb{N}$ is considered as a term, it has its own type $\mathbb{N} : \mathcal{U}$ where $\mathcal{U}$ is a universe of types so that $\mathcal{U}$ is closed under forming functions, products, sums etc. When $\mathcal{U}$ is considered as a term, it has its own type $\mathcal{U} : \mathcal{U}_1$ and $\mathcal{U}_1 : \mathcal{U}_2$ etc. This hierarchy of universes is cumulative i.e. $\alpha : \mathcal{U}_i$ means that $\alpha : \mathcal{U}_j$ for all $i \leq j$.[2] The reason for this hierarchy of universes is because if $\mathcal{U} : \mathcal{U}$ is allowed, then the system would be inconsistent because of the Girard paradox.[3] The intuition behind Girard paradox is that universe is an extensional type so that it is determined up to what is contained as a term of that universe; consequently a universe $\mathcal{U} : \mathcal{U}$ would imply a never completable construction

[1]Although the term "term" has an unwanted linguistic connotation, but the term "term" is standard.

[2]the subscript i, j should not be confused with terms of natural numbers, they are intended as labels only and a bigger subscript means a "bigger" universe. Indeed $n \mapsto \mathcal{U}_n$ is ill-typed because this "function" has to lie in a universe $\mathcal{V}$ of all universes which is inconsistent because in particular $\mathcal{V} : \mathcal{V}$ is ill-formed.

of $\mathcal{U}$ similar to how the "set" of all set is problematic in set theory.[3]

Type and proposition Following Ladyman, a type can be considered as a concept while a term of a type can be understood as witness of the concept so that we do not have to enter the debate of ontological status of mathematical objects.[6] Under this interpretation, for example the type $\mathbb{N}$ is understood as the concept of natural number and $1 : \mathbb{N}$ is as a specific instance of the concept natural number. Whether or not the natural number exists in any significant sense at all, we can nonetheless form the concept of natural number, indeed otherwise we would not be able to even ask the question of existence of natural number. Then any type α can be thought as a proposition, namely the proposition that α has a term and by exhibiting a term $a : \alpha$ we prove the proposition corresponding to α, although most types contains much more information other than that its inhabitedness, that is, there might be different terms witnessing the same type. This is where HoTT diverges from classical logic, because there are non-equivalent ways to prove the same proposition and this makes a huge difference, see section 4. In order to practice mathematics in HoTT, one at least need to be able to form conjunction, disjunction and material implication of any two propositions, negation of any proposition and universally and exisitentially quantified propositions; this will be achieved under Curry-Howard correspondence. A quick overview is given in Appendix A. In what follows, unless said otherwise Greek letters will be denoting types and corresponding lower case Latin letter with possible subscripts denoting its term (so for example $a : \alpha$) and upper case Latin letter denoting its corresponding proposition à la Curry-Howard (so for example A is the proposition that α has term) and mathcal font letters denoting universes (although universes are types as well). In the next few paragraphs, some non-classical behaviours of HoTT are exhibited.

Currying Functions often take more than one inputs to give a result. In HoTT, this can be done in two ways. One is by taking the product type of all inputs so that the product type would contain all the informations for the said multiple inputs; this is often called an uncurried function. Alternatively, a multiple-input function can be achieved by currying, i.e. instead of taking all the input simultaneously, one creates a function which takes multiple

[3] for everything after this point, the level of universe is not been dealt carefully, thus everything involve a universe should be read with a typical ambiguity, i.e. replace it with higher level universe if necessary.

inputs one by one. For example, to define $(x, y) \mapsto x^y$, one instead defines $x \mapsto (y \mapsto x^y)$; that is, to define exponentiation, one defines a function which upon receiving an input x gives a function computing powers of x. Curried functions are more common practice in HoTT. This seems to be unnatural to common mathematical practice, but currying is often useful in common practice as well, for example natural number addition can be implemented by fixing one addend first or that Riesz representation theorem is proved by fixing one input of an inner product.

Total functions There are no partial functions in HoTT, if a function f is of type $\alpha \to \beta$, then for every $a : \alpha$, $f(a) : \beta$ and thus, in particular, must be defined. If one really need to define a partial function g, one form a type $\beta_\perp$ by taking the coproduct of β and $\mathbf{1}$ so that if $g(a)$ is not defined, $g(a)$ is $\perp$, see section A.4.

Divergence from classical logic It is unheard of that double negation elimination is not present in an intuitionistic logic, but in HoTT, double negation elimination cannot even be consistently assumed. Indeed, there is a type α such that there is a term of type of type $(((\alpha \to \mathbf{0}) \to \mathbf{0}) \to \alpha) \to \mathbf{0}$ witnessing the proposition that $\neg(\neg\neg A \implies A)$ provided that α represents A. Since double negation elimination and the law of excluded middle are materially equivalent, the law of excluded middle cannot be consistently assumed as well. The crude justification for the invalidity of is a demand to know exactly which disjunct is true, however the law of excluded middle would not give such information, see also section 4. Similarly, from "$\neg\forall$" to "$\exists\neg$" cannot be assumed as well since it is implied by double negation elimination. Thus, one cannot in general construct a term $\left(\left(\prod_{a:\alpha} \beta(a)\right) \to \mathbf{0}\right) \to \sum_{a:\alpha} (\beta(a) \to \mathbf{0})$. This because even we know that $\left(\prod_{a:\alpha} \beta(a)\right) \to \mathbf{0}$, there is in general not a way to construct term $a : \alpha$ enabling us to construct a term of type $\sum_\alpha \beta$. Dependent product can also be used to encode subtype, for a term of type $\sum_\alpha \beta$ are of the form (a, b) such that $a : \alpha$ and $b : \beta(a)$, i.e. a is such that $\beta(a)$ holds. When using dependent pair to encode subsets we in general will not necessarily have $\left(A^\complement\right)^\complement = A$ where $\cdot^\complement$ denotes set theoretic complement because of issues of double negation elimination. Also, x can occur in $\{a \in A \mid B(a)\}$ more than once, this is because we might be able to construct (a, b_1) and (a, b_2) such that $b_1 \neq_{\beta(a)} b_2$, i.e. we can witness $\beta(a)$ in more than one non-identical ways. This is the proof-relevant feature of HoTT that there might be many ways to prove a proposition and we are able to compare the different proofs, see also section 3.1.

3 Identity Type

In this section, identity type is introduced in section 3.1 exactly as presented in HoTT Book, similarly for the univalent axiom. Then I will briefly explain how to form identity across different types in HoTT. In the rest of this section, I will try to give a justification of path induction.

3.1 Overview

Identity type over a type $\alpha : \mathcal{U}$, written as $\cdot =_\alpha \cdot : \mathcal{U}$, has a single constructor $\mathrm{refl}_\alpha : \prod_{a:\alpha} a =_\alpha a$. If context is clear, it is common practice to write refl_a to mean $\mathrm{refl}_\alpha(a)$. This corresponds to that, given $a, a' : \alpha$, one could always ask if $a = a'$, but only knows $a = a$ without any more information. Given a family of types $\gamma : \prod_{a_1,a_2:\alpha} a_1 =_\alpha a_2 \to \mathcal{U}$ and a $c : \gamma(a, a, \mathrm{refl_a})$, then a function of type $\prod_{a_1,a_2:\alpha} \prod_{p:a_1=_\alpha a_2} \gamma(a_1, a_2, p)$ can be constructed such that $f(a, a, \mathrm{refl_a}) \equiv c$. More formally we have:

$$\mathrm{ind}_{\cdot =_\alpha \cdot} : \prod_{\gamma:\prod_{a_1,a_2:\alpha} a_1=_\alpha a_2 \to \mathcal{U}} \left(\prod_{a:\alpha} \gamma(a, a, \mathrm{refl}_a) \right) \to \left(\prod_{a_1,a_2:\alpha} \prod_{p:a_1=_\alpha a_2} \gamma(a_1, a_2, p) \right)$$

such that $\mathrm{ind}_{\cdot =_\alpha \cdot}(\gamma, c, a, a, \mathrm{refl}_a) \equiv c(a)$. The inductor states that to construct a term from identity type, it is sufficient to work out the "recipe" only for trivial self identities of a for all term $a : \alpha$. We often find the inductor for type $a =_\alpha \cdot$ more convenient, where $a : \alpha$ is a fixed term. The inductor for this type states that to construct a thing from identity whose left hand side is a, it is sufficient to work out the "recipe" only for refl_a. More formally:

$$\mathrm{ind}_{a =_\alpha \cdot} : \prod_{\gamma:\prod_{a':\alpha}(a=_\alpha a') \to \mathcal{U}} \gamma(a, \mathrm{refl}_a) \to \left(\prod_{a':\alpha} \prod_{p:a=_\alpha a'} \gamma(a', p) \right)$$

such that $\mathrm{ind}_{a =_\alpha \cdot}(\gamma, c, a, \mathrm{refl}_a) \equiv c$. In fact, $\mathrm{ind}_{\cdot =_\alpha \cdot}$ and $\mathrm{ind}_{a =_\alpha \cdot}$ are logically equivalent, thus I will call them both path induction.

Proof relevancy Assume that identity type deserves its name in this paragraph. Since identity type can be formed over any type, for any $a, a' : \alpha$ and $p, q : a =_\alpha a'$, one can form type $p =_{a=_\alpha a'} q$; and for any $x, y : p =_{a=_\alpha a'} q$, one can form the type $x =_{p=_{a=_\alpha a'} q} y$, etc. These types forms an infinite groupoid structure, this is one of many reasons that HoTT itself is of mathematical interest. Under Curry-Howard correspondence, whether or not this type is inhabited corresponds to whether or not the two witnesses of a being identical with a' are equal. Moreover given any proposition A, assume α formalises A in HoTT and $a, a' : \alpha$ formalises two proofs of A, then it makes sense to compare a and a'. Though the notion of equality of proofs may sound to be fine-grained, it may not be the case for some types. Just when two proofs are considered to be equal by HoTT is a difficult question to answer, this is part of reasons why the ∞-groupoid structure of identity type is mathematically sophisticated. For example, $\mathbb{N}$ has decidable equality, i.e. if $p_1, p_2 : a = b$ where $a, b : \mathbb{N}$, then p_1 and p_2 are guaranteed to be equal in the sense of HoTT; but for other types, identity type might not be decidable.

Judgemental equality Judgemental equality, denoted by $\equiv$, holds between two terms if and only if two terms are the same by definition or more formally the same after a series of α, β, η reduction/conversions. To emphasise the difference between judgemental equality and identity type, the latter is often called propositional identity. Judgemental equality *cannot* be formalised inside HoTT, i.e. $a \equiv b$ is not a valid expression in HoTT while propositional equality $a = b$ (when the types of a, b are the same) is a valid expression in HoTT. Since judgemental equality is "equal by definition", we use $:\equiv$ to introduce a definition.

3.1.1 Minimality and functoriality

Assume path induction in this section. In this section, we show that identity type is at least not outrageously wrong as identity in ordinary sense. We note that identity is necessarily the minimal equivalent relation that respects all function application. We show that identity type is both minimal and functorial. Informally, a minimal equivalent relation R is an equivalent relation such that, for any equivalent relation R', aRb implies $aR'b$; a functorial relation R is such that for any function f, aRb implies $f(a)Rf(b)$.

Minimality Identity type is minimal. Since $aR'b$ would automatically hold as long as a equals b and R' is an equivalent relation, we need to prove that identity type is reflexive, symmetric and transitive.

Reflexivity This is just the constructor, refl.

Symmetry We want a term of type $\gamma(a, a', p) :\equiv \left(a' =_\alpha a\right)$ where $\gamma : \prod_{a,a':\alpha} \left(a =_\alpha a'\right) \to \mathcal{U}$, by path induction, we only need to construct a term of type $\gamma(a, a, \mathrm{refl}_a)$ — we use refl_a. We also write this as $\cdot^{-1}$.

Transitivity We want a term of type $\gamma(a_1, a_2, p) :\equiv \prod_{\substack{a_3:\alpha \\ q:a_2=_\alpha a_3}} a_1 =_\alpha a_3$, by path induction we only need to construct a term of type $\gamma(a, a, \mathrm{refl}_a) \equiv \prod_{\substack{a':\alpha \\ q:a=_\alpha a'}} a =_\alpha a'$ — we use $a' \mapsto \mathbb{1}_{a=_\alpha a'}$. We also write this as $\cdot \centerdot \cdot$.

Functoriality Identity type is functorial. We need to show that for any $f : \alpha \to \beta$ and $a =_\alpha a'$, $f(a) =_\beta f(a')$, i.e. we want a term of type $\gamma(a, a', p) :\equiv f(a) =_\beta f(a')$, by path induction, we only need a term of type $\gamma(a, a, \mathrm{refl}_a) \equiv f(a) =_\beta f(a)$ — we use $\mathrm{refl}_{f(a)}$. We also write this as $\mathrm{ap}_f(p)$

However, this does not justify identity type of its name for this relies on path induction and that the only requirement for identi ty is minimality and functorality. The former is not intuitive and we move justification of path induction to latter sections; and the latter condition is only necessary but not sufficient.

3.1.2 Transport

In this section we show indiscernible of identicals. Let β be a family of types indexed by α and $p : a =_\alpha a'$. Then $\beta(a) \to \beta(a')$ is witnessed. By path induction, we only need to show that $\beta(a) \to \beta(a)$ is witnessed, indeed it is witnessed by $\mathbb{1}_{\beta(a)}$. Instead of call this function indiscernible of identities, we call it $\mathtt{transport}^\beta_p : \beta(a) \to \beta(a')$, because this function will transport a term of type $\beta(a)$ to a term of type $\beta(a')$ according to p. By how `transport` is defined and path induction, we have $\mathtt{transport}^\beta_{\mathrm{refl}_a} \equiv \mathbb{1}_{\beta(a)}$.

3.1.3 Identity system

An identity system ι over α is a family of types $\alpha \to (\alpha \to \mathcal{U})$ with the same behaviour as the identity type on α. This is:

- we have a term $\rho : \prod_{a:\alpha} \iota(a, a)$. (This is like the refl : $\prod_{a:\alpha} a =_\alpha a$.)
- Given any type family $\gamma : \prod_{a,a':\alpha} \iota(a, a') \to \mathcal{U}$ and a term $c : \prod_{a:\alpha} \gamma(a, a, \rho(a))$, we can construct a term $f : \prod_{a,a':\alpha} \prod_{p:\iota(a,a')} \gamma(a, a', p)$ such that $f(a, a, \rho(a)) = c(a)$. (So ρ has the same induction principle as identity except that computation only hold propositionally instead of judgmentally.)

It turns out that ρ is an identity system over α is logically equivalent to that for any $a : \alpha$, the type $\sum_{a':\alpha} \iota(a, a')$ is contractible, i.e. it has a specified term and all other terms are propositionally equal to the specified term.

3.2 Univalent Axiom

The univalent axiom very roughly states that isomorphism is isomorphic to identity:

$$(\alpha \simeq \beta) \simeq (\alpha = \beta).$$

In this section we will make this slogan precise. Then we will utilise this section to build an account of identity in HoTT with the univalent axiom in the next section. Since univalent axiom involves "$\simeq$", we will make this notion precise first.

3.2.1 Isomorphism, Equivalence and Univalence

A function $f : \alpha \to \beta$ is an isomorphism when it is a structure preserving invertible map, that is if there another structure preserving function $g : \beta \to \alpha$ such that $g \circ f \sim \mathbb{1}_\alpha$ and $f \circ g \sim \mathbb{1}_\beta$, where $\cdot \sim \cdot$ denotes that two functions are pointwise equal. We use "$\sim$" instead of "=" because without univalent axiom, function extensionality is not provable. In HoTT, any invertible map would be an isomorphism because functions in HoTT are automatically structure preserving. To properly implement a function between structures, the structural information must already be present in the type of domain and codomain. For example the type of groups in universe $\mathcal{U}$ can be defined as $\texttt{Group} :\equiv \sum_{\alpha:\mathcal{U}} \texttt{GroupStructure}(\alpha)$ where $\texttt{GroupStructure}(\alpha)$ encodes a binary associative operation, a neutral element and an unary inverse operation. Then if we have two groups

γ_1, γ_2 : `Group`, then a function $f : \gamma_1 \to \gamma_2$ must provide a recipe to produce $\mathrm{pr}_2(\gamma_2)$: `GroupStructure`$(\mathrm{pr}_1(\gamma_1))$ from the underlying "set" of γ_1 and its group structure. If one truly wants to define a function that does not respect structures, one should define a function of type $\gamma_1 \to \mathrm{pr}_1(\gamma_2)$ instead. So we can define the type of isomorphism between α and β to be $\texttt{Iso}(\alpha, \beta) :\equiv \sum_{f:\alpha\to\beta} \texttt{isIsomorphism(f)}$ where $\texttt{isIsomorphism}(f)$ is defined as $\sum_{g:\beta\to\gamma} (g \circ f \sim \mathbb{1}_\alpha) \times (f \circ g \sim \mathbb{1}_\beta)$. However, if "$\cdot \simeq \cdot$" is taken to be $\texttt{Iso}(\cdot, \cdot)$, the univalent axiom would be inconsistent. This is because `Iso` contains more information than the classical notion of isomorphism, namely there might be non-equivalent way of proving a function being an isomorphism. The obvious solution is use proposition truncation, and this would serve a functioning univalent axiom.[4] A more constructive approach is to use bi-invertibility $\texttt{Biinvert} :\equiv \sum_{f:\alpha\to\beta} \texttt{isBiinvertible}(f)$ where $\texttt{isBiinvertible}(f) :\equiv \left(\sum_{g:\beta\to\alpha} g \circ f \sim \mathbb{1}_\alpha\right) \times \left(\sum_{h:\beta\to\alpha} f \circ g \sim \mathbb{1}_\beta\right)$, in fact $\texttt{isIsomorphism}(f)$ and $\texttt{isBinvertible}(f)$ are logically equivalent, but `Biinvert` does not contain any more information other than itsbi-invertibility. Thus the slogan should be "Equivalence is equivalent to identity". This clears the definition of $\cdot \simeq \cdot$, next we will formally introduce univalent axiom.

Since universe $\mathcal{U}$ is a type, we have identity type $\cdot =_\mathcal{U} \cdot$ of $\mathcal{U}$. Then we can define a function by path induction $\mathrm{id2equiv}(\alpha, \beta) : \alpha =_\mathcal{U} \beta \to \alpha \simeq \beta$ such that $\mathrm{id2equiv}(\alpha, \beta, \mathrm{refl}_\alpha) :\equiv ((\mathbb{1}_\alpha, (\mathbb{1}_\alpha, a \mapsto \mathrm{refl}_a)), (\mathbb{1}_\alpha, (\mathbb{1}_\alpha, a \mapsto \mathrm{refl}_a)))$. Note that $\mathrm{id2equiv} \equiv \texttt{transport}^{\alpha\mapsto\alpha}((\mathbb{1}_\alpha, a \mapsto \mathrm{refl}_a))$, i.e. the equivalence given by id2equiv from a path $p : \alpha =_\mathcal{U} \beta$ is given by moving the equivalence $\mathbb{1}_\alpha$ between α and α to an equivalence between α and β. The univalent axiom is the statement that id2equiv is an equivalence. We write $\mathrm{ua} : \alpha \simeq \beta \to \alpha =_\mathcal{U} \beta$ to be the inverse of id2equiv. So the univalent axiom not only postulates the existence of $\mathrm{ua} : \alpha \simeq \beta \to \alpha =_\mathcal{U} \beta$, it further asserts that ua and id2equiv are inverses, we always have $\mathrm{pr}_1(\mathrm{id2equiv}(\mathrm{ua}(e)))(a) =_\beta \mathrm{pr}_1(e)(a)$ for all equivalence $e : \alpha \simeq \beta$. Since being an equivalence is a mere proposition, we will use an equivalence as if it were a function, for example we write $\mathrm{pr}_1(e)(a)$ simply as $e(a)$. Because id2equiv is an equivalence, we also

[4]See section 4.

have that all equalities between types come from some equivalence between types. So we can perform induction on equivalences as well: given a family of types $\gamma : \alpha \simeq \beta \to \mathcal{U}$, by univalent axiom we have $\gamma' : \alpha = \beta \to \mathcal{U}$ such that $\gamma = \gamma' \circ \text{id2equiv}$. Then to prove $\prod_{e:\alpha\simeq\beta} \gamma(e)$, we can prove that $\prod_{p:\alpha=\beta} \gamma'(p)$. By path induction, we need only a proof of $\gamma'(\text{refl}_\alpha)$ which corresponds to a proof of $\gamma((\mathbb{1}_\alpha, \_))$ where $\_$ is any term witnessing that $\mathbb{1}_\alpha$ is a equivalence.[5] As a sanity check, note that under the univalent, we are never identifying any discernible types, for otherwise these types cannot be equivalent.

3.3 Heterogeneous identity

An immediate objection for identity type presented in section 3.1 is that the supposed identity is only relative to a type α, i.e. if $a : \alpha$ and $b : \beta$ are of different type then it is not even legitimate to ask if $a = b$. Whether the natural number 0 equals the real number 0 or not, one should at least be able to talk about this. I present how heterogeneous identity can be defined but I also argue why it is not generally useful in HoTT plus the univalent axiom.

Heterogeneous identity Assume $\alpha : \mathcal{U}_i$ and $\beta : \mathcal{U}_j$. One can define an inductive type $\cdot \overset{\mathbf{H}}{\underset{\alpha,\beta}{=}} \cdot : \mathcal{U}_{\max(i,j)}$ with only one constructor hrefl : $\prod_{a:\alpha} \left(a \overset{\mathbf{H}}{\underset{\alpha,\alpha}{=}} a \right)$.[6] I do not attempt to justify that heterogeneous identity deserves its name of identity, for, after explaining its purpose in this paragraph, I will argue that all its purpose can be fulfilled by identity type plus univalent axiom and, in latter section, I will attempt to justify that identity type and path induction. Heterogeneous identity is to circumvent dependent types. For example $\beta : \alpha \to \mathcal{U}$ and $f : \prod_{a:\alpha} \beta(a)$, then given $a =_\alpha a'$, we can show $f(a) \overset{\mathbf{H}}{\underset{\alpha,\alpha}{=}} f(a')$. In particular since $2 = 2 + 0$, $\mathbb{R}^2 \overset{\mathbf{H}}{\underset{\mathcal{U},\mathcal{U}}{=}} \mathbb{R}^{2+0}$. However I think this heterogeneous identity type is unsatisfying because the resulting type must live in a bigger universe than α and β for otherwise it is inconsistent while identity type lives in the same universe as α. So if heterogeneous identity type were to replace identity type in HoTT, there would not be an infinite groupoid structure of heterogeneous identity type, because there would not

[5] since being equivalence is a mere proposition, it does not matter which term is used.

[6] This is also called John Major equality, it is part of standard library of Coq under the name `JMeq`.

be universe large enough to contain it, and thus HoTT would lose part of its mathematical interest. Also heterogeneous identity type is not really a generalisation of identity type, for even though $a =_\alpha a'$ implies $a \underset{\alpha,\alpha}{\overset{\mathbf{H}}{=}} a'$, the converse is not true due to proof relevancy of HoTT. More specifically, heterogenous identity between $a : \alpha$ and $b : \beta$ is materially equivalent to identity between (α, a) and (β, b) where the latter is not materially equivalent to the identity between a and b. Because comparison of (α, a) and (β, b) in HoTT is dependent on how α and β are identified in HoTT sense under univalent axiom.

I argue the correct way to compare two terms of different type is to use a sigma type so that their types are remembered. So to compare whether $a : \alpha : \mathcal{U}$ is equal to $b : \beta : \mathcal{U}$, one form the type $(\alpha, a) =_{\sum_{\mathcal{U}:\mathcal{U}'} \mathcal{U}} (\beta, b)$. By transporting, to construct a term of $(\alpha, a) =_{\sum_{\mathcal{U}:\mathcal{U}'} \mathcal{U}} (\beta, b)$ is equivalent to construct a term $p : \alpha = \beta$ and then construct a term of type $\mathtt{transport}_p^{\mathbb{1}_\mathcal{U}}(a) =_\beta b$.[7] In this sense, a natural number 0 is definitely not equal to the real number 0 if we assume univalent axiom. For otherwise the natural number is equal to real number and, by univalent axiom, there is a bijection between them. I do not think this necessarily goes against mathematical Platonism, for I will later develop a philosophically neutral reading of identity type and, under this reading, the claim that $(\mathbb{N}, 0 : \mathbb{N}) \neq (\mathbb{R}, 0 : \mathbb{R})$ is only the claim that not all proof involving $0 : \mathbb{N}$ can be "transported" into a corresponding proof involving $0 : \mathbb{R}$. It is perhaps interesting to observe that since $\mathbb{N}$ and $\mathbb{Q}$ are equinumerous, it is at least possible to have $\mathbb{N} = \mathbb{Q}$ via univalent axiom, thus it is at least possible to have $(\mathbb{N}, 0 : \mathbb{N}) = (\mathbb{Q}, 0 : \mathbb{Q})$, however, if this is true, it does not assert that $0 : \mathbb{N}$ and $0 : \mathbb{Q}$ are the same in a Platonistic sense under my reading, it only assert that a proof involving $0 : \mathbb{N}$ can be transported into a corresponding proof involving $0 : \mathbb{R}$ according to how natural number and rational number are identified. I also remark that this is not the most outrageous practice to only be able to make equality claim by remembering extra type information, consider the following example. There is a clear sense in which $0 \neq_\mathbb{R} 1$ holds. However since $\mathtt{AddGrp} :\equiv (\mathbb{R}, +, 0)$ and $\mathtt{MultGrp} :\equiv (\mathbb{R}^+, \times, 1)$ are isomorphic (via exp) and hence equal under univalent axiom, $(\mathtt{AddGrp}, 0 : |\mathtt{AddGrp}|) = (\mathtt{MultGrp}, 1 : |\mathtt{MultGrp}|)$ also makes (a homotopy type theoretic) sense; colloquially, it is the claim that 0 and 1 serves the same (group theoretic) role in two isomorphic groups. In this example, since the most ambient identity is equality as real numbers, no

[7] The technical details are quite long, see [13, theorem 2.7.2].

mathematician would seriously claim that $0 = 1$ without writing a remark. But when there is not an immediate sense in which two things could be compared, a mathematician perhaps would write down "flip of a pentagon equals to $(0, 1) \in \mathbb{Z}/5\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$". This is not claiming identity should hold between "flip" and $(0, 1)$; rather, I only claim that HoTT can make sense of (and even make it rigorous) what sometimes mathematicians informally and abusively equates.

Whether or not the identity type in HoTT in any sense resembles the true identity relation is not my main concern, for I aim to develop a neutral reading without appealing to identity type encoding identity so that the resulted HoTT can be appreciated by mathematicians with different philosophy of mathematics standpoint.

3.4 A justification of path induction

Even if we ignore the issue of the inability of forming identity type across different types, the induction principle for identity type is still highly unintuitive. It states that to prove a theorem about all identities of terms of type α, it suffices to prove the theorem only for trivial self-identities. Essentially, this is like doing mathematical induction on natural number with only base case and call it a day. Ladyman justifies induction principle by appealing to uniqueness principle of identity type and substitution salva veritate.[4] I think the argument is unsatisfactory in at least two ways. Firstly, uniqueness principle of all other types are derived based on the corresponding induction principles, it makes the justification in some degree ad hoc if the uniqueness principle of identity type must be assumed. Moreover, the uniqueness principle of identity type states that by fixing $a_1 : \alpha$, then for any $a_2 : \alpha$ and $p : a_1 = a_2$, one would have a term of type $(a_1, \text{refl}_{a_1}) =_{\sum_{a':a}(a=_\alpha a')} (a_2, p)$ which itself could not be justified by intuition. Secondly, substitution salva veritate can only be assumed if one assumes that identity type is or closely resembles the real identity relation. But if one can justify this resemblence, the induction principle of identity type should be accepted as or at least approximation of induction on the real identity relation in some sense via Curry-Howard correspondence. I think to justify identity type as an approximation of the true identity relation is a much harder task. Thus, I propose an alternative route — without assuming identity type codifies the real identity relation, I argue that its induction principles can still be justified without appealing to what we think about the real identity relation. And then use the justified induction principle of identity type to give a neutral reading of HoTT.

Since this section is to discuss induction principle of identity type, it would be unjustified of me to call it identity type when what it is is unclear. Thus, we use $a\boxed{\alpha}b$ to denote identity type with constructor `box` : $\prod_{a:\alpha} a\boxed{\alpha}a$ in HoTT and just call it box type in order not to be deceived by the symbol $\cdot = . \cdot$. Whenever $a\boxed{\alpha}b$, we say a $\alpha-$boxes b and if context permits, just a boxes b. The induction principle for box type is

$$\mathrm{ind}_{\cdot\boxed{\alpha}\cdot} : \prod_{\gamma:\prod_{a_1,a_2:\alpha} a_1\boxed{\alpha}a_2 \to \mathcal{U}} \left(\prod_{a:\alpha} \gamma(a, a, \mathrm{refl}_a)\right) \to \left(\prod_{a_1,a_2:\alpha} \prod_{p:a_1\boxed{\alpha}a_2} \gamma(a_1, a_2, p)\right).$$

The induction principle of box type is justified by box type being an instance of W-type. And W-type is justified by its usefulness, for example, natural numbers, lists are all instances of W-type. As long as we do not give box induction principle any philosophical significance, let us tolerate box induction by not assuming any meaning except the type of box induction. Then, correspondingly, one can define a box system to be a family of types $\beta : \alpha \to (\alpha \to \mathcal{U})$ such that:

1. we have a term $\rho : \prod_{a:\alpha} \beta(a, a)$
2. Given any type family $\gamma : \prod_{a,a':\alpha} \beta(a, a') \to \mathcal{W}$ and a term $c : \prod_{a:\alpha} \gamma(a, a, \rho(a))$, we can construct a term $f : \prod_{\substack{a,a':\alpha \\ p:\beta(a,a')}} \gamma(a, a', p)$ such that $f(a, a, \rho(a))\Box c(a)$.

A box system is of course logically equivalent to that for any $a : \alpha$, the type $\sum_{a':\alpha} \beta(a, a')$ is box-contractible, i.e. it has a specified term and all other term boxes the specified term.

Suppose we work with universes up to $\mathcal{W}$. We now construct a box system over α. Consider the following type

$$\iota(a, a') :\equiv \left(\sum_{\delta:\alpha\to\mathcal{W}} \big((\delta(a) \leftrightarrow \delta(a')) \to \mathbf{0}\big)\right) \to \mathbf{0}.$$

This type states that the existence of a δ such that δ discerns a from a' would be contradictory, i.e. ι is (a version of) indiscernibility. We now prove that ι is a box system by proving it is box-contractible. Because $\mathbf{0}$ is a type such that every terms of $\mathbf{0}$ boxes each other, we have that $\iota(a, a')$ is also a type such that every terms of $\iota(a, a')$ boxes each other. So we only need to exhibit a term of $\sum_{a':\alpha} \iota(a, a')$ — we use $i :\equiv \left(a, (\delta, \mathrm{H}_\delta) \mapsto \mathrm{H}_\delta \left(\mathbb{1}_{\delta(a)}, \mathbb{1}_{\delta(a)}\right)\right)$. We always have $i : \iota(a, a)$ defined as above. And we can then define a box_1 system to be a family of types $\beta : \alpha \to (\alpha \to \mathcal{U})$ such that

1. we have a term $\rho : \prod_{a:\alpha} \beta(a, a)$
2. Given any type family $\gamma : \prod_{a,a':\alpha} \beta(a, a') \to \mathcal{W}$ and a term $c : \prod_{a:\alpha} \gamma(a, a, \rho(a))$, we can construct a term $f : \prod_{\substack{a,a':\alpha \\ p:\beta(a,a')}} \gamma(a, a', p)$ such that $\iota(f(a, a, \rho(a)), c(a))$.

Again, box_1 system is logically equivalent to $\sum_{a':\alpha} \beta(a, a')$ is box_1-contractible; using this, one can prove again that ι is a box_1 system with "constructor" $i : \prod_{a:\alpha} \iota(a, a)$. Then the second clause of definition of box_1 system states that given any type family $\gamma : \prod_{a,a':\alpha} \iota(a, a') \to \mathcal{W}$ and a term $c : \prod_{a:\alpha} \gamma(a, a, i(a))$, one can construct a term $f : \prod_{\substack{a,a':\alpha \\ p:\iota(a,a')}} \gamma(a, a', p)$ such that $\iota(f(a, a, i(a)), c(a))$. To put symbols in words, for any predicate γ involving terms of α being indiscernible, if we always know that γ holds for all terms of α which are discernible from itself in a trivial way, then we know that γ holds for all indiscernible terms, trivially indiscernible or non-trivially indiscernible. Let us call this indiscernibility induction. Before we continue with justifying path induction, let us note that $a \Box b$ implies $\iota(a, b)$.

We now show a variant of uniqueness principle for box type. Given $a, a' : \alpha$ and $p : a \boxed{\alpha} a'$, we have $\iota\left((a, \mathtt{box}_a), (a', p)\right)$. To see this, we need to assume that $(\delta, \mathrm{H}_\delta) : \sum_{\delta} \left(\delta(a, \mathtt{box}_a) \leftrightarrow \delta(a', p)\right) \to \mathbf{0}$ and derive a contradiction. However by a box induction, one get $(a, \mathtt{box}_a) \Box (a', p)$ and hence

$\iota((a, \texttt{box}_a), (a', p))$ and thus a contradiction follows. Then we use indiscernibility induction to derive substitution salva veritate for ι, i.e. we want to show that for any a, a' such that $\iota(a, a')$ and any family of types γ, we would have $\gamma(a) \to \gamma(a')$. By indiscernibility induction, we only need to show this for a and $i : \iota(a, a)$, then we take $\mathbb{1}_{\gamma(a)}$. Now with the variant of uniqueness principle and substitution salva veritate, we could derive box induction again: fix a $\gamma : \prod_{a':\alpha}(a' \Box a) \to \mathcal{W}$ and a term $c : \gamma(a, \texttt{box}_a)$, we want a term of type $\prod_{\substack{a' \\ p:a'\Box a}} \gamma(a', p)$. Fix a' and p, by substitution salva veritate, we only need to construct a term of $\gamma(a, \texttt{box}_a)$ since the variant of uniqueness principle tell us $\iota((a, \texttt{box}_a), (a', p))$; then we can just take c.

Let me summarise my argument:

1. Based on usefulness of W-type, we tolerate the existence of box type on programatic ground, but we do not assume any meaning of it;
2. without assuming any meaning of box induction, we derived indiscernibility induction;
3. by indiscernibility induction, we derive a variant of uniqueness principle for box type and substitution salva veritate for indiscernibility type ι;
4. using the variant of uniqueness principle and substitution salva veritate, we infer box induction again.

On the first sight, this looks like a huge effort to derive a tautology — by assuming box induction, we infer box induction. From step 1 to step 2, we do not assume box type to bear any meaning other than that box induction has a particular type, which we do not assume to have any philosophically significance as well. But ι indeed has meaning, namely it convey a version of indiscernibility. Its meaning is granted just by its type under Curry-Howard correspondence alone. And ι has an "induction" principle of the same shape as box induction. Since ι has meaning, its induction principle has meaning as well, namely that any theorem/construction for indiscernible terms can be sufficiently proved/constructed only on trivially self-indiscernible terms. Then using indiscernibility induction, we can give meaning to a variant of uniqueness principle that any two "boxes" are indiscernible in the sense of ι and indiscernibility satisfies substitution salva veritate. Then finally, we get box induction again, but now box induction is derived from meaningful indiscernibility induction and thus carries weight. So in a nutshell, box induction is justified because as long as we know that any two “boxes” are indiscernible and indiscernible terms can be substituted salva veritate then a proof of box induction can be provided. And in this proof, we did not utilise any assumption of the real identity relation being encoded by box types.

The first objection/confusion I anticipate is the transition from *meaningless* box induction to *meaningful* indiscernibility induction and it is implausible to have meaning suddenly appearing. The meaning of indiscernibility induction is from type information of ι under Curry-Howard correspondence, not from box induction. I think an analogy of auxiliary line in plane geometry might help. For example one wants to prove that D is mid point of AB in a triangle ABC under some condition, and it so happens that an auxiliary line from C to D helps the proof. The auxiliary line can help with proofs not because the auxiliary line is assumed to be median line, it just make information easier to see; and probably a proof without using auxiliary line exists. Similarly, there perhaps are proofs of indiscernibility induction without involving box type and box induction, but box type and box induction just makes the work easier.

The next objection is that ι is not indiscernibility. I can think of two reasons:

1. ι at best is indiscernibility up to universe $\mathcal{W}$.
2. The negation of ι does not give us an actual way to discern objects because of nonexistence of a double negation elimination law.

The second point is more of a debate between classical mathematics vs. intuitionism, and it is too big a topic to be contained in this paragraph so we move it to later sections. For the first point, I will start my defence by noting $\mathcal{W}$ is arbitrary and the above argument would work in any universe and universes are closed under (dependent) product, coproduct, (dependent) function, box and W type, so if we have decided which universe our mathematical object of study lives in, then that universe tends to be mathematically rich enough. Thus, even if we only stay in a universe, there are a lot to be proved. And if it is indeed necessary to investigate in some even larger universe, and since universes are assumed to be cumulative, ι can be moved into the larger universe. Admittedly, not every interesting question can be asked if we assume a maximal universe even if the maximal universe can be arbitrarily large. For example, highly likely, it is inconsistent to formalise HoTT with unlimited universe hierarchy within HoTT with a maximal universe. The defect of ι is that it cannot quantify over every possible properties/predicates/functions whatsoever. But if there is any framework that can do this without a Russell or Girard flavoured paradox, then that framework is certainly better in this perspective. I doubt if there is any.

Another objection/question is why anyone should tolerate W-type in the first place. As mentioned above, if one can prove indiscernibility induction without using box induction, then box induction would be justified without appealing to W type and would be justified solely on ι. Unfortunately, I am not able to offer such proof. Let me at least try to get away with it. W-type is certainly a useful generalisation whether its existence is philosophically problematic or not. One can justifiably question why anyone should believe in identity type on the basis of its unintuitive induction principle. But I neither took box type to encode identity, nor box induction to encode path induction. Box induction is a result of box type being an instance of W-type and I only use it to prove indiscernibility induction. Think it as an auxiliary line, nowhere did I claim box induction exists or being real, it is at most used as a tool. The box induction principle in the conclusion of the argument is based on indiscernibility induction whose meaning is significant due to Curry-Howard correspondence, *not* due to box induction. If my reader pays close attention, she will notice that the above argument also relies on that $a \Box b$ implies $\iota(a, b)$ and thus she can accuse me of sneakily using "box type is identity". The accusation would make sense if only something thing close to identity can imply indiscernibility up to $\mathcal{W}$. But then I do not even need to bother with steps 3 and 4 because I would have something even better — box type encodes identity. Also, I do not think that only something resembling identity can imply indiscernibility when indiscernibility is only up to a scope of information. For example, $(\mathbb{R}, |\cdot|)$ and $(\mathbb{R}, 2|\cdot|)$ are topologically indistinguishable but different as metric space. If this example is not convincing because metric spaces and topological spaces can always coexists at the same universe level, consider the smallest universe consisting $\mathbf{0}$, then any two types in this universe are equivalent hence box each other under univalent axiom and hence are indiscernible; but are they all *really* the same, for example even $\iota(\mathbf{0} \times \mathbf{0}, \mathbf{0})$ in this barren universe only freely generated by $\mathbf{0}$, it is still controversial to claim that $\mathbf{0} \times \mathbf{0}$ and $\mathbf{0}$ are the same type.

My reader may also note that the box induction obtained as the result of the above argument is not path induction verbatim. Given a type family γ indexed by identity/box type and $c : \prod_a \gamma(a, \mathrm{refl}_a)$, path induction should give us a function f such that $f(a, \mathrm{refl}_a) \equiv c(a)$, but box induction as we argued only gives us $\iota(f(a, \mathrm{refl}_a), c(a))$. I do not think this is a problem for me because in subsequent sections, I would only use the box/path induction to define functions without using the judgemental equality $f(a, \mathrm{refl}_a) \equiv c(a)$.

To end this section, I stress that the whole argument is to justify box induction using indiscernibility; the argument is not about whether or not box type encodes or approximates identity. Thus one can freely decide whether identity type is identity without contradicting the above argument.

3.5 Identity type and univalent axiom only as transporter

In this section I develop a neutral account of box/identity type barring the differences between classical mathematics and intuitionism for now. To avoid unintentionally use intuition about identity, I will continue to use box notations.

Assuming that box induction is justified, the transport function can be defined according to section 3.1.2. For any type family β indexed by α and a box $p : a \Box a'$, the type of $\mathtt{transport}^{\beta}_{p}$ is $\beta(a) \to \beta(a')$ and the type of $\mathtt{transport}^{\beta}_{p^{-1}}$ is $\beta(a') \to \beta(a)$. The transport function states that if a boxes a', then given a proof of $\beta(a)$, not only do we know that $\beta(a')$, but we have a corresponding proof of it. With univalent axiom, given a proof of a theorem about α, not only do we know a corresponding theorem of β, we actually know a proof of β as long as α and β are equivalent. My reading of box type is that the main purpose, if not the only purpose, of box type is to define functions like `transport` and ap. These function with univalent axiom would provide an instrument for mathematicians no matter what their positions of philosophy of mathematics are as long as their positions can make sense of Curry-Howard correspondence. Then the principal worry is that since I have not given box type any meaning at all, why one should believe that `transport` actually preserves proofs.

Given the success of HoTT translating a large chunk of mathematics, for all the positions of philosophy of mathematics in which a proof in set theoretic foundation is acceptable at face value or can somehow be transcribed to be acceptable, a corresponding proof in HoTT can be accepted as face value or can somehow be transcribed to be acceptable. This claim is not radical, because it is essentially claiming that HoTT is a sufficiently capable language to write mathematics and this can be demonstrated by many mathematical theories already formalised in HoTT. Then, independent of one's position of philosophy of mathematics, she should accept a term of type α corresponds to a proof of some proposition encoded by α in HoTT because of Curry-Howard correspondence. Thus, independent of one's position of philosophy of mathematics and what box type actually is, as long as box induction is justified, the information conveyed in the type of `transport` under Curry-Howard correspondence defined using box induction should be accepted. And the type information under Curry-Howard correspondence, tell us that `transport` indeed transport one proof to another. Similarly for the univalent axiom, since all it does it to turn an equivalence between types into boxes between types, univalent axiom under this reading should not be understood as equating or identifying types, it is at most relating types by boxes. Thus by not giving box type any specific meaning, one can enjoy the convenience of univalent axiom without worrying univalent axiom's claim between identity and equivalence. Since in this reading, box/identity type is not given any meaning and hence does not necessarily encode true identity, I should explain how this reading can make sense of uniqueness claim which requires identity to make sense of. For example, under this reading, the group theoretic claim that there is one and only one identity element in a group would become there is a specified term of a group such that all other group satisfying such and such property will box that specified term. This sound nothing like a uniqueness claim for $0 + 0$ and 0 are the same identity element because $0 + 0 = 0$ but merely that $0 + 0 \Box 0$ does not convince me that they should be counted as the same, and indeed that is the point of box notation. I propose the solution to be that $0 + 0 \Box 0$ implies $\iota(0 + 0, 0)$ so that a uniqueness claim should be understood as a claim of indiscernibility up to a maximal universe. I have argued that $\Box$ implying ι does not necessarily force $\Box$ to convey meaning similar to identity and, even if $\Box$ implying ι indeed forces $\Box$ to be somewhat like the real identity, it would be to my advantage in section 3.4. Similarly $a \leq b$ would be interpreted as that $a +$ something is indiscernible from b in the sense of ι. Then one might ask, why we should bother with box type anyway, it sounds like ι does all the work. The first reason is because box type is needed (at least before I find a better proof) to justify indiscernibility induction. The second reason is because ι has a fixed meaning but $\Box$ in this reading does not, so if one actually thinks that $\Box$ encodes identity, she is free to adopt so but she may or may not think ι as identity; so box type gives a little bit more wiggle room for interpretation.

Similarly, the univalent axiom should also not be interpreted as identifying or equating types in any strong sense. I am not claiming that equivalence introduced in section 3.2 is without meaning. Equivalence is encoded by bi-invertibility which asserts a function f with functions g, h such that $g \circ f(a)$ always boxes a and $f \circ h(b)$ always boxes b. Given boxes as proof transport interpretation, equivalence asserts functions f, g, h such that a proof involving $g \circ f(a)$ can always be transported to a corresponding proof involving a and a proof involving $f \circ h(b)$ can always be transported to a corresponding proof involving b. Univalent axiom then asserts that equivalence is equivalent to box type. Even though under this reading, the univalent axiom does not make any significant claim between types, it still gives a function `ua` which takes an equivalence and produces a box. We are just not giving any meaning to the box produced by `ua`. Given an equivalence e between α and β, $\mathtt{ua}(e)$ is a box such that proofs about α can be transported to corresponding proofs about β. This is coherent with box type only as proof transporter: an equivalence e from α to β gives us the ability to transport proofs about a term of α to proofs about a term of β pointwise; and $\mathtt{ua}(e)$ gives us the ability to transport proofs about α to proofs of β globally.

Since I am developing an instrumentalist account of HoTT, I will exhibit why one should believe proofs in the sense of HoTT corresponds to or is not very different from our pre-theoretic notion of proof. A pre-theoretic notion of proof is a series of steps of reasoning from some premises where each step follows from some justifiable or intuitive rules. A proof in HoTT is a term of some type and terms are built by applying constructor, elimination and induction rules and functions to already built terms. Most constructor and elimination rules (induction rules respectively) behave like logical rules (induction principle for the corresponding mathematical objects) under Curry-Howard correspondence so that a term can be interpreted as applying different logical rules sequentially as well. Since HoTT is often implemented in a computer language, a proof in HoTT often appears to be very different from a proof in ordinary sense, but the difference is often only in appearances — a proof written in a foreign script is a proof nonetheless. Under my reading of HoTT, a motivation to use HoTT is that it automatically transports a proof of a theorem about some type to a proof of a theorem about all other equivalent types, i.e. we get theorems and proofs for free. But one might argue that in other foundation like ZFC, theorems can also be transported, for example cyclicity of group is preserved under group isomorphism. I think there are at least two advantages of HoTT over ZFC in these cases:

1. In ZFC, it is not always clear when properties are structural or preserved under isomorphism. Consider again $(\mathbb{R}, +, 0)$ and $(\mathbb{R}^+, \times, 1)$, even though they are isomorphic as groups, their (most obvious) topology is different. In HoTT, if a proof of theorem about topology of $(\mathbb{R}, +, 0)$ is transported to $(\mathbb{R}^+, \times, 1)$, the proof and theorem would be automatically about the topology induced by the group isomorphism. Of course, HoTT will not prevent anyone from making these mistakes, the claim is that `transport` in HoTT is a rigorously defined proof transporter while "preserved by iso/homeo/diffeo/...morphism" is a term tending to be glossed over. If one is to transport a proof about topology of the additive group, the transported proof would not be about the default topology of the multiplicative group but about the corresponding topology transported to the multiplicative group. This is not to say ZFC must be inferior in this perspective for these mistakes/ignorance (if anyone is making them) are made by us and not intrinsic to ZFC. Since HoTT is often implemented as a computer system, if anyone is indeed transporting a non-structural property, HoTT would complain. However, HoTT is not magical in the sense that it would save us from ever mistaking a structural property from a non-structural one, for if a property is structural, then its structuralness is often part of definition and hence a proof must be provided if it is formalised properly. This will only sounds cumbersome to mathematicians, but for a careless ZFC practitioner, this will save her from thinking that a theorem about the standard topology of a multiplicative group is proved because a similar theorem is proved for the standard topology of a additive group even if the two groups are group theoretically isomorphic. Thus, in this respect, HoTT is often correctly being accused that all trivial things has to be proved. As a defence, if it is indeed trivial, it would not be very difficult to write it down anyway and most computer version HoTT has some automation built-in.

2. `transport` can also transport constructions due to proof relevancy.

> For example, `transport` not only tells us deffeomorphic manifolds have the same cohomology, it also transports a sequence computing cohomology of one manifold to another sequence computing that of the other deffeomorphic one. Thus not only two deffeomorphic manifolds has the same cohomology, if we have a sequence to compute one cohomology, we *automatically* get a corresponding sequence via transport. Again, I am not claiming that ZFC foundation is not able to do this, these kind of practice is often only quickly mentioned under the phrase "similarly". Admittedly, sometimes the only purpose of a construction is to prove the theorem and once the theorem is proved, constructions used in the proof tend to be forgotten in these cases. And if one unfolds the construction thus `transport`ed, it would just be applying the structural preserving map and/or its inverse to the original construction which is what one would expect "similarly" to mean anyway. Thus, one can object that even if `transport` can transport constructions, its usefulness is still doubtful. Still, I think even if constructions are ad hoc, free ad hoc constructions are still not a harmful product and `transport` gives a systematic meaning of "similarly", namely if one unfold the `transport`ed construction, she will see how similar they are.

Another motivation to use HoTT is higher inductive types which can often help with constructions via quotients and thus saving proofs of well-definedness. I will use construction of integer and integer addition as an example. A higher inductive type is an inductive type with some pre-determined boxes/paths. For example, integer in HoTT can be defined as a type with two constructor $\texttt{pos} : \mathbb{N} \to \mathbb{Z}$ and $\texttt{neg} : \mathbb{N} \to \mathbb{Z}$ and a box/path $z : \texttt{pos}(0)\boxed{\mathbb{Z}}\texttt{neg}(0)$. Then the subtraction function $\cdot - \cdot : \mathbb{N} \to \mathbb{N} \to \mathbb{Z}$ can be defined via pattern matching by $0 - n :\equiv \texttt{neg}(n)$, $n - 0 :\equiv \texttt{pos}(n)$ and $\texttt{succ}(n) - \texttt{succ}(m) :\equiv n - m$. To check this definition makes sense, one need to check this definition respects the pre-determined box z, i.e. $0 - 0 \equiv \texttt{pos}(0)$ and $0 - 0 \equiv \texttt{neg}(0)$ but $z : \texttt{pos}(0)\boxed{\mathbb{Z}}\texttt{neg}(0)$. Then integer addition $\cdot + \cdot : \mathbb{Z} \to \mathbb{Z} \to \mathbb{Z}$ can be defined by $\texttt{pos}(n) + \texttt{pos}(m) :\equiv \texttt{pos}(n +_{\mathbb{N}} m)$, $\texttt{neg}(n) + \texttt{neg}(m) :\equiv \texttt{neg}(n +_{\mathbb{N}} m)$, $\texttt{pos}(n) + \texttt{neg}(m) :\equiv n - m$ and $\texttt{neg}(n) + \texttt{pos}(m) :\equiv m - n$. To see this definition makes sense, one needs to check that $\texttt{pos}(0) + \texttt{pos}(0)\boxed{\mathbb{Z}}\texttt{pos}(0) + \texttt{neg}(0)\boxed{\mathbb{Z}}\texttt{neg}(0) + \texttt{pos}(0)\boxed{\mathbb{Z}}\texttt{neg}(0) + \texttt{neg}(0)\boxed{\mathbb{Z}}$, and all boxes are just true by definitions of $\cdot + \cdot$ and/or $\cdot - \cdot$ and/or $z : \texttt{pos}(0)\boxed{\mathbb{Z}}\texttt{neg}(0)$. All the checks of well-definedness are almost mindless, and if one is defining this on a computer HoTT, these are automatic. And this does not cost the intuitiveness of definition — integers has two copy of natural numbers, a positive copy and a negative copy such that both zeros are equal while integer defined as a higher inductive type encodes exactly this except that both zeros box each other hence are indiscernible in the sense of ι. In contrast, in ZFC, $\mathbb{Z}$ can be defined as $\mathbb{N} \times \mathbb{N}/\sim$ where $(n_1, m_1) \sim (n_2, m_2) \iff n_1 + m_2 = n_2 + m_1$ and $[(n_1, m_1)]_\sim + [(n_2, m_2)]_\sim :\equiv [(n_1 + n_2, m_1 + m_2)]_\sim$. To check this definition makes sense, one need to check

- $\sim$ is an equivalent relation;
- if $(n_1, m_1) \sim (n_1', m_1')$ then $[(n_1', m_1')]_\sim + [(n_2, m_2)]_\sim = [(n_1, m_1)]_\sim + [(n_2, m_2)]_\sim$;
- if $(n_2, m_2) \sim (n_2', m_2')$ then $[(n_1, m_1)]_\sim + [(n_2', m_2')]_\sim = [(n_1, m_1)]_\sim + [(n_2, m_2)]_\sim$.

To check these are by no means hard, but it is not as mindless, because commutativity and associativity has to be invoked now. Perhaps there are clever way to encode integer in ZFC such that well-definedness check would be more mindless, but that would not be as natural as $\mathbb{N} \times \mathbb{N}/\sim$. One can complain that this is not really fair to ZFC theorists for in ZFC, integers can also be defined axiomatically as the structure such that there are two injections from $\mathbb{N}$ such that the two injection agrees on 0. The only reason that ZFC theorist constructs integers the harder way is because a concrete representation of integer can be obtained via the harder way. However, ZFC theorist must construct a concrete representation of integers because only then she will know that the axiomatic definition is not void/inconsistent. However, all higher inductive types can be justified to be consistent systematically albeit the justification is much harder than construction of integer. Thus, as long as an axiomatic description can be written as a higher inductive type, we would know the description is consistent. This includes more than integers, higher inductive types can also encode shapes naturally, for example the circle S^1 is a type with a single constructor $o : \mathsf{S}^1$ and a box/path `circle` : $o \Box o$. Note that without the box `circle`, S^1 is the unit type. So the possibility of non-trivial boxes/paths gives HoTT's unique ability of defining higher inductive types. Other examples include doughnut shapes etc. Thus even if ZFC theorists can provide an axiomatic definition for each shape, they need a concrete representation in each case so that they can know that the shape is not void. Of course, I am not claiming these concrete representation is hard, the point is that as long as one accepts the consistency of higher inductive type, the consistency of the description captured by a higher inductive type is automatic.

Now, I need to justify higher inductive types are compatible with my reading of box/identity type as only proof transporter. The main worry is that boxes/paths postulated when defining a higher inductive types must carry meaning, otherwise the higher inductive types could not encode the desired mathematical object, for example $z : \texttt{pos}(0)\Box\texttt{neg}(0)$ must convey something like that the "positive zero" and "negative zero" are the same so that the integer type actually encodes the mathematical object integers. Like before, when we do need a meaning to be attached to boxes/paths, we opt to use indiscernibility up to $\mathcal{W}$ implied by boxes/paths. So `circle` postulates that other than the indiscernibility implied by $i : o\Box o$, there is another sense in which o is indiscernible from itself, namely `circle`.

To summarise this section, once box induction can be justified without appealing to box encodes or approximates identity, we could define transport function without appealing to meaning of box types. I argue that since the way in which a term is built in HoTT resembles our concept of proof, under Curry-Howard correspondence, the transport function indeed transports one proof or construction to a corresponding proof or construction. This provides a motivation of using HoTT, not because ZFC does not have "proof transporter", but because transport function in HoTT is rigorously defined and, since HoTT is often implemented on a computer, one cannot "transport wrong properties", for example transport a topological property over groups. Another motivation is higher inductive types, i.e. inductive types with some specified boxes/paths. Higher inductive types can make the process of checking well-definedness easier or even automatic; and higher inductive types can be used to encode shapes. Since throughout the argument, box types are never given any meaning, when one indeed need an interpretation or explanation of box type to justify that a type can convey some information, she could use indiscernibility (up to $\mathcal{W}$) implied by box type.

Another point is missing, HoTT is constructive but mathematical practice, at least classically, is not. To claim this account provides a neutral tool, I need to either argue that constructivism being forced upon is a feature not a bug, or that my account of HoTT is compatible with traditional mathematical practice in some sense. In the next section, I choose the second route.

HoTT without true identity Under my reading, in HoTT, there would be no true identity type but only box type without any significant meaning. This does not concerns me because I am proposing to view HoTT as an instrument, at least for mathematicians who can accept Curry-Howard correspondence. If I am successful, HoTT under my reading should not force any ontological commitment upon its user and any mathematician using HoTT should be able to retain their own notion of identity.

4 Can HoTT be made compatible with classical mathematics?

Constructive mathematics, unlike classical mathematics requires proof to provide a recipe of construction; for example a proof of existential claim should give us a method of actually producing a mathematical entity with some desired properties; a proof of disjunctive claim should enable us to actually determine which of the disjunct is true. Thus, the law of excluded middle, or equivalently the law of double negation elimination and axiom of choice are not assumed in constructive mathematics. Because of this, many mathematicians are reluctant to practice constructivism and thus reluctant to adopt HoTT or any other variants of Martin-Löf type theory as a mathematical framework. I do not plan to advocate constructivism because not all classical theorems are constructively provable. Since I advocate an instrumentalist account of HoTT which mathematicians with different philosophy of mathematics view can enjoy, I will argue that HoTT can provided a framework similar to a classical one by showing how to incorporate the law of excluded middle. The axiom of choice can be dealt with the same techniques. But first, I will take detour to show that constructivism is not absolutely undesirable either.

A famous example that is often cited to show the necessity of law of excluded middle is to prove existence of a rational irrational power. The proof is the following: Let $a = \sqrt{2}^{\sqrt{2}}$, either a is rational or it is irrational; if it is rational then there is nothing left to prove; if it is irrational, then $a^{\sqrt{2}} = \sqrt{2}^{2} = 2$ is a rational irrational power. I have two problem with this example. Firstly, law of excluded middle is not indispensable in this example, because we can in fact prove constructively that $\sqrt{2}^{\sqrt{2}}$ is transcendental and hence irrational via Gelfond-Schneider theorem, albeit the proof (constructive version or classical version) of Gelfond-Schneider theorem is much harder than the proof in the example. Thus, law of excluded middle in this example just make the example a lot easier, without it, a proof is still possible. Secondly, there is even an easy constructive proof of existence of rational irrational power. $\sqrt{2}$ is irrational and $2\log_2 3$ is irrational.[8] Then $\sqrt{2}^{2\log_2 3} = 2^{\frac{1}{2}\cdot 2\log_2 3} = 3$. My objections of this example is not that the law of excluded middle is not important or should not be assumed. I only refuse to believe this example can show the importance of the law of excluded middle. Now let us compare these two proofs, the first proof essentially shows that (assuming the law of excluded middle), either $\sqrt{2}^{\sqrt{2}^{\sqrt{2}}}$ is rational or $\sqrt{2}^{\sqrt{2}}$ is rational but we still have no idea which one is actually rational; while the second proof shows that $\sqrt{2}^{2\log_2 3}$ is rational. Although they both show existence of a rational irrational power in a classical sense, the fist proof will not actually give us a rational irrational power while the second proof does. I think the difference between these two proofs illustrates that constructive proofs are not always undesirable.

Vanilla version of HoTT is indeed constructive because existential claim (disjunctive claim respectively) is encoded by dependent pair type (coproduct type respectively). Law of excluded middle in HoTT is encoded by

$$\mathsf{LEM} :\equiv \prod_{\alpha:\mathcal{U}} \alpha + (\alpha \to \mathbf{0}).$$

Not only is LEM not provable in HoTT, it is actually inconsistent. A proof of this fact utilises that there is a non-trivial box $\mathbf{2}\square\mathbf{2}$ brought by the univalent axiom. One obvious response is to simply abandon the univalent axiom and assume LEM as an axiom instead. But for me, this is not a great solution for at least two reasons. Firstly, even without the univalent axiom, I am not sure if the rest of HoTT is actually compatible with LEM.[9] Secondly, consider approach (a) in which we remove the univalent axiom and assume LEM assuming that LEM can be consistently added to HoTT in this way, would it not be a better solution to assume the univalent axiom while restricting LEM to only some types as long as those LEM-applicable types are rich enough

[8] Otherwise, $\log_2 3 = \frac{a}{b}$, then $3^b = 2^a$ but 3^b is odd and (because obviously $a \neq 0$) 2^a is even, a contradiction. This proof does not involve double negation elimination and it is constructively acceptable.

[9] I think/guess higher inductive type might be a potential problem.

to write mathematics in? Call the latter approach (b). I think the approach (b) is better because without the univalent axiom, we could not transport proofs among equivalent types and axiom of function extensionality needs to be assumed as well. Under box type and the univalent axiom only as proof transporter reading, the univalent axiom is not a philosophically loaded axiom to be assumed for it is not assumed to identify or equate any types. On the other hand, since there is a substantial debate about constructive mathematics and constructive logic in particular, the law of excluded middle must be at least a somewhat controversial assumption. Thus if by assuming the univalent axiom and a restricted version of the law of excluded middle provided that the restricted version is still rich enough, we can recover all mathematical practice that is implementable in HoTT with LEM but without the univalent axiom, then we are achieving the same result with (at most) the same cost. The expensive part in both ways is the law of excluded middle but arguably the restricted version is cheaper since it is restricted. The only concern is that by restricting the law of excluded middle, more assumptions and/or constructions are needed to write mathematics. This is not the case because the law of excluded middle can be consistently assumed for mere propositions and mere propositions can be implemented by a higher inductive type whose requirement is box type, common to both approaches (a) and (b). In the following, I will explain what mere propositions are and why they are sufficient to write mathematics.

If α satisfies the condition that for any $a, a' : \alpha$, we have a term of $a \Box a'$, then assuming $\alpha + (\alpha \to \mathbf{0})$ would not lead to inconsistency. Such types are called mere propositions. Being a mere proposition is formalisable inside HoTT, it can be encoded as

$$\texttt{isProp}(\alpha) :\equiv \prod_{a,a':\alpha} a \Box a';$$

and $\texttt{isProp}$ is itself a mere proposition. The restricted version of the law of excluded middle can be encoded as

$$\mathsf{LEM}_{\mathsf{r}} :\equiv \prod_{\alpha:\mathcal{U}} \texttt{isProp}(\alpha) \to (\alpha + (\alpha \to \mathbf{0}))\,.$$

One immediate curiosity/objection against mere proposition under my reading is "why should behaviour of box type be able to affect whether or not law of excluded middle should be applied, after all box types are supposed to not carry any meaning." This is again because box type implies indiscernibility up to $\mathcal{W}$. Recall that the proof of LEM being inconsistent with HoTT plus the univalent axiom utilises a nontrivial box $\mathbf{2}\Box\mathbf{2}$, this nontrivial box implies that $\mathbf{2}$ is indiscernible from $\mathbf{2}$ even if one views $0 : \mathbf{2}$ as $1 : \mathbf{2}$ and $1 : \mathbf{2}$ as $0 : \mathbf{2}$. Under my reading, box type does not, in itself, affect when one can apply law of excluded middle, its implication, namely indiscernibility leads to inconsistency of the law of excluded middle.

Another objection is that there are essentially only two mere propositions, $\mathbf{0}$ and $\mathbf{1}$. Here, I take "essentially" to mean something like that for any mere proposition ϕ, either $\phi \simeq \mathbf{0}$ or $\phi \simeq \mathbf{1}$ and hence under the univalent axiom, either $\phi\Box\mathbf{0}$ or $\phi\Box\mathbf{1}$. If this is what "essentially" means, I do not have problem with this claim, because $\phi\Box\mathbf{0}$ means a proof of ϕ can be transported to a proof of $\mathbf{0}$ and vice versa so that ϕ is not provable; $\phi\Box\mathbf{1}$ means that firstly ϕ is provable because a term of $\mathbf{1}$ can be transported back to ϕ and secondly any proof of ϕ can be transported to $\mathbf{1}$. The second point is really just a tautology for from any type α, there is always $\alpha \to \mathbf{1}$, so $\phi\Box\mathbf{1}$ just asserts ϕ is provable. So mere propositions are the types that contains no more information than inhabitedness/provability. Perhaps it is strange to think that $\phi\Box\psi$ whenever they are both provable or both unprovable mere propositions. But under my reading, the strangeness can be eased — $\phi\Box\psi$ means a proof of ϕ can be transported to a corresponding proof of ψ and (since also $\psi\Box\phi$) a proof of ψ can be transported to a corresponding proof of ϕ; and this is indeed true, any proof of ϕ can be transported to a proof of ψ because ψ is provable and one can just take that proof of ψ and vice versa. This is of course to saying that all mere provable mere propositions are trivially provable by transporting $\star : \mathbf{1}$, because a box/path is needed in order to use transport and, in this case, the box/path is from an equivalence under univalent axiom and equivalence between a mere proposition and $\mathbf{1}$ is not trivial. Similarly, unprovable mere propositions are not trivially unprovable as well. To make it easier to accept the notion of $\mathtt{isProp}(\alpha)$, one can think it as the truth value of the proposition corresponding to α, then if that proposition is true, $\mathtt{isProp}(\alpha)$ is witnessed; otherwise, unwitnessed. And since, if we only care about whether the proposition corresponding to α is true or not, it would not matter how its truth is witnessed, thus all terms (if any) of $\mathtt{isProp}(\alpha)$ would not make a difference. Next we demonstrate that mere propositions are rich enough to write mathematics.

For any type α, we can form the type $\|\alpha\|$ via higher inductive types with constructor $|\cdot| : \alpha \to \|\alpha\|$ and boxes $x\Box y$ for any $x, y : \|\alpha\|$. Then proposition truncation is rich enough to write mathematics in a classical sense. To illustrate this point, let us assume LEM_r and that we truncate everywhere, most notably $\left\|\sum_{a:\alpha}\beta(a)\right\|$ and $\|\alpha+\beta\|$ to mean "there exists an a such that $\beta(a)$" and "α or β", then all classical logic rules can be recovered, thanks to the induction principle of proposition truncation which states that for any type γ such that $b \Box b'$ for all $b, b' : \beta$, for any function $g : \alpha \to \beta$, there is a function $\hat{g} : \|\alpha\| \to \beta$ such that $\hat{g}(|a|) \equiv g(a)$ and an auxiliary property about boxes that we do not really need in this essay. For example, law of excluded middle will be true by assumption and double negation elimination would also be true because, in HoTT, it is logically equivalent to the law of excluded middle and that a "not exists" claim is equivalent to "for all not" is also provable in a truncated sense. In classical mathematics, if one wants to prove β from a premise of an existential claim, then one can freely postulated an entity warranted by that existential claim. If a dependent pair is untruncated then $\mathtt{pr}_1$ will also give us an entity to use. But if dependent pair is truncated, there is no longer a $\mathtt{pr}_1$ available. But proposition truncation can mimic the classical behaviour of existential claim as long as β is also a mere proposition, because by the induction principle of propositional truncation, to prove that β is implied by the truncated existential claim, it is sufficient to prove that β is implied by the untruncated existential claim. However if β is not a mere proposition, then the induction principle is not very useful. This should not be very surprising, because if β is not a mere proposition, then the type information of β contains more than whether or not β is inhabited/provable, for example if β is an untruncated existential claim (i.e. a dependent pair), then to prove β, one actually need to produce an entity and a proof of why that entity has the desired properties. Then generally, a mere existential claim would not contain enough information, namely a concrete entity to start with, to produce a concrete entity that β requires. This is not to say to prove an untruncated claim from a truncated claim is never possible, it is just that the induction principle of proposition truncation will no longer be helpful.

The last paragraph should be seen as an invitation to truncate everywhere. Remember that the introduction of proposition truncation is to make the law of excluded middle and axiom of choice compatible. However, as I have argued in the example of rational irrational power, the law of excluded middle is not always necessary and by avoiding the law of excluded middle, proofs in the end can sometimes be more informative; thus, to truncate everything and everywhere is often unnecessarily cutting informations away. The same is true for the law of double negation elimination, for example instead of assuming a non-emptiness condition, just assume an inhabitedness condition. But constructivism is not forced upon to any user of HoTT, it is at most a suggestion; if anyone wants to truncate everywhere then she is free to truncate everywhere. However, I want to point out that mathematical practice is more like a classical approach mixed with at least some occasional constructivism. Here are some examples. Most existential claim in mathematical practice are actually untruncated, for instance, most proofs of existential claims are actually recipes to produce the desired entity; this approach can be very accurately captured by dependent pairs. Most claims with disjunctions are better described as a coproduct type, because most of time we indeed know which disjunct is true. And it is not very unusual for mathematician to carefully state that her proof involves axiom of choice or its consequences so that, with care, constructive part can be separated from classical counterpart.

5 Conclusion

In this essay, I develop an alternative neutral reading of HoTT so that mathematicians with philosophy of mathematics standpoint other than structuralism can use. To achieve this, I argue to not interpret identity type as encoding or approximating identity; instead, view it as meaningless, uninterpreted type from which one can derive indiscernibility induction. Using indiscernibility induction, one can justify a variant of uniqueness principle of (uninterpreted, meaningless) identity type and substitution salva veritate. Combining these two lemmas, the induction principle of identity type (path induction) can be justified. Then using path induction, I develop the account of identity type and univalence axiom merely as only proof transporter.

A Logic in HoTT

A.1 Function, Dependent Function, Implication, Universal Quantifier

Function type If $\alpha : \mathcal{U}_i, \beta : \mathcal{U}_j$ are two types, then there is a type called function from α to β denoted by $\alpha \to \beta : \mathcal{U}_{\max(i,j)}$, i.e. the type so formed lives in a universe large enough to contain both α and β. In this case, α is referred to as the domain and β as the codomain. A term of this type is given by $a \mapsto b$ where a is a variable occurring in b so that whenever every occurrence of a in b is replaced with a term $x : \alpha$, the resulting expression is a term of β. This process is denoted as $(a \mapsto b)(x)$, or $f(x)$ if the function $a \mapsto b$ is given a name f, this process is also called the eliminator of function type. Any function f is judgmentally equal to $x \mapsto f(x)$ and any two functions are judgmentally equal if they are alpha-equivalent, i.e. all the same except for variable renaming without clashing, for example $x \mapsto x + 1 : \mathbb{N} \to \mathbb{N}$ and $y \mapsto y + 1 : \mathbb{N} \to \mathbb{N}$ are judgmentally. Judgemental equality is denoted by $\cdot \equiv \cdot$. Judgemental equality is equality by definition. Since domains and codomains can be any type, they can be function types as well. This provides us with the facility of defining functions with more than one inputs — $f :\equiv \alpha \to (\beta \to \gamma)$ is a function which upon receiving $a : \alpha$ gives another function $f_a :\equiv f(a) : \beta \to \gamma$.

On the logical side, function type behaves like material implication. To prove $A \implies B$ is to prove B under the assumption of A. By previous paragraph, this is equivalent to constructing a function $\mathrm{imp}_{A,B} : \alpha \to \beta$ because $\mathrm{imp}_{A,B}$ upon receiving any term $a : \alpha$ gives a term $\mathrm{imp}_{A,B}(a) : \beta$, i.e. given any a as a witness or proof of proposition A, $\mathrm{imp}_{A,B}(a)$ is a witness or proof of proposition B. This is the $\implies$ introduction rule in logic. The eliminator of function type serves as modus ponus.

Dependent function type The codomain of a function type is always fixed, thus dependent function is introduced as a generalisation of function type. The codomain of a function type will vary according to its input. A family of type indexed by α is a term $\beta : \alpha \to \mathcal{U}$, i.e. for any term $a : \alpha$, $\beta(a) : \mathcal{U}$ is a type. For α and β, the dependent function type $\prod_{(a:\alpha)} \beta(a)$ (or $\prod_{\alpha} \beta$) can be formed.[10] A term of $\prod_{(a:\alpha)} \beta(a)$ is of the form $a \mapsto b$ where a is a dummy variable contained in b such that after every occurrence of a is replaced with a term $x : \alpha$, the results would be a term of type $\beta(x)$. Exactly like function type, for any dependent function f, $f \equiv a \mapsto f(a)$ and two dependent functions are judgmentally equal if they are alpha-equivalent.

[10]alternatively, one can write this as $(a : \alpha) \to \beta(a)$, this notation reminds one that the behaviour of dependent function type is similar to that of function type and, in particular, if β is a constant family, the notion of dependent function type collapse into ordinary function type.

On the logical side, β can be thought as a family of proposition indexed by what corresponds to α. A term f of type $\prod_{(a:A)} \beta(a)$ corresponds to a proof of $\forall a : A, B(a)$, because for any $a : \alpha$, $f(a)$ witness $\beta(a)$ hence corresponds to a proof of $B(a)$. This corresponds to universal quantifier introduction rule. For elimination rule, just observe that if under the assumption that $\forall a : A, B(a)$ and an $a : A$, one can assume a term $f : \prod_{a:\alpha} \beta(a)$ and $a : \alpha$ hence $f(a)$ corresponds to the proposition $B(a)$.

A.2 Zero, Negation

The zero type (denoted as $\mathbf{0}$) is an instance of inductive types in HoTT.[11] To specify an inductive type, constructors, eliminators and if any, computation rule and uniqueness rule need to be specified. If α is an inductively-defined type, its constructors are to specify how its terms are constructed and eliminators are to specify how a function from α can be introduced. Computation rules are to specify how constructors and eliminators are related. In the case of $\mathbf{0}$ there is no constructors so that no terms of type $\mathbf{0}$ can be constructed. The elimination rule states that given any type α, there is a function $!_\alpha : \mathbf{0} \to \alpha$.

On the logical side, $\mathbf{0}$ corresponds to the "canonical" false proposition. This is because false proposition should not be witnessed (at least not consistently). The eliminator of $\mathbf{0}$ is *ex falso* — any proposition A is derivable from a false proposition, $!_\alpha : \mathbf{0} \to \alpha$ witness this. Then $\alpha \to \mathbf{0}$ corresponds to the proposition $\neg A$.

A.3 Product, Dependent Pair, Conjunction, Existential quantifier

Product of α and β is defined inductively with a constructor $(\cdot\,,\cdot) : \alpha \to (\beta \to \alpha \times \beta)$, i.e. given any $a : \alpha, b : \beta$, one has $(a, b) : \alpha \times \beta$. The first elimination rule (or the recursor) is as following: to construct a term of type $f : \alpha \times \beta \to \gamma$, it suffices to construct $g : \alpha \to \beta \to \gamma$ such that $f((a, b)) \equiv g(a)(b)$. More formally, there is $\mathrm{rec}_{\alpha\times\beta} : \prod_{\gamma:\mathcal{U}} (\alpha \to (\beta \to \gamma)) \to (\alpha \times \beta \to \gamma)$ such that $\mathrm{rec}_{\alpha\times\beta}(\gamma)(g) :\equiv ((a, b) \mapsto g(a)(b))$. For example, one

[11] this essay will not cover the general syntax of inductive type and its variant in any detail, for details see for example chapter 5 of [13]

can form $\mathrm{pr}_1 :\equiv \mathrm{rec}_{\alpha\times\beta}(\alpha)(a \mapsto (b \mapsto a)) : \alpha \times \beta \to \alpha$, and similarly $\mathrm{pr}_2 :\equiv \mathrm{rec}_{\alpha\times\beta}(\beta)(a \mapsto \mathbb{1}_\beta)$, we often invoke recursor implicitly to define a function via "pattern matching" $(a, b) \mapsto \cdots$. This is sometimes called uncurrying a function. The other elimination rule (or the inductor) states that if $\gamma : \alpha \times \beta \to \mathcal{U}$ is a family of types then to construct a dependent function $\prod_{x:\alpha\times\beta} \gamma(x)$, it suffices to construct a dependent function $\prod_{a:\alpha}\prod_{b:\beta} \gamma((a, b))$. More formally,

$$\mathrm{ind}_{\alpha\times\beta} : \prod_{\gamma:\alpha\times\beta\to\mathcal{U}} \left(\prod_{a:\alpha}\prod_{b:\beta} \gamma((a, b))\right) \to \left(\prod_{x:\alpha\times\beta} \gamma(x)\right),$$

such that $\mathrm{ind}_{\alpha\times\beta}(\gamma)(g)((a, b)) :\equiv g(a)(b)$. Note that if γ is a constant family, then the inductor collapse into the recursor. Similar to the recursor, we can implicitly using the inductor to define functions via pattern matching $(a, b) \mapsto \cdots$. By this eliminator, in the rest of this paper, the notions of $f((\cdot\,, \cdot))$ and $f(\cdot)(\cdot)$ would be dealt not with care anymore, we simply write $f(\cdot\,, \cdot)$.

On the logical side, product $\alpha \times \beta$ corresponds to $A \wedge B$, because to prove $A \wedge B$, one provides a proof of A and provides a proof of B; this is equivalent to provide a term $a : \alpha$ and $b : \beta$ and or equivalently (via constructor $(\cdot\,, \cdot)$ and functions $\mathrm{pr}_1, \mathrm{pr}_2$) a term $(a, b) : \alpha \times \beta$. Then the constructor $(\cdot\,, \cdot)$ corresponds to conjunction introduction rule while pr_1 (respectively pr_2) corresponds to conjunction left (respectively right) elimination rule.

Dependent pair[12] For a family of type β indexed by α, the product type can be generalised to dependent pair type denoted by $\sum_{(a:\alpha)} \beta(a)$ or $\sum_\alpha \beta$ [13].

The constructor is

$$(\cdot\,, \cdot) : \prod_{(a:\alpha)} \prod_{(b:\beta(a))} \sum_{(a:\alpha)} \beta(a),$$

[12] we do not call it dependent product because dependent product often refers to dependent function because of the $\prod$ sign in its notation, nor do we call it dependent sum despite the $\sum$ sign in the notation because sum type is another kind of type, see section A.4.

[13] alternatively, one can write it as $(a : \alpha) \times \beta(a)$ or even, provided clear context, $\alpha \times \beta$ to remind the similarity between product type and dependent pair

that is, for any term $a : \alpha$ and term $b : \beta(a)$, a term $(a, b) : \sum_{a:\alpha} \beta(a)$ can be constructed. The corresponding recursor is

$$\mathrm{rec}_{\sum_\alpha \beta} : \prod_{\gamma:\mathcal{U}} \left(\prod_{(x:\alpha)} \beta(x) \to \gamma \right) \to \left(\left(\sum_{(x:\alpha)} \beta(x) \right) \to \gamma \right)$$

such that $\mathrm{rec}_{\sum_\alpha \beta}(\gamma, g, (a, b)) :\equiv g(a)(b)$. Or one could invoke the eliminator implicitly by defining a function via pattern matching $(a, b) \mapsto \cdots$. For example $\mathrm{pr}_1((a, b)) :\equiv a : \left(\sum_\alpha \beta \right) \to \alpha$, but to define pr_2, we need the inductor:

$$\mathrm{ind}_{\sum_\alpha \beta} : \prod_{\gamma:\sum_\alpha \beta \to \mathcal{U}} \left(\prod_{a:\alpha} \prod_{b:\beta(a)} \gamma((a, b)) \right) \to \prod_{x:\sum_\alpha \beta} \gamma(x)$$

such that $\mathrm{ind}_{\sum_\alpha \beta}(\gamma, g, (a, b)) \equiv g(a)(b)$. We want $\mathrm{pr}_2 : \prod_{(x:\sum_\alpha \beta)} \beta(\mathrm{pr}_1(x))$, thus $\mathrm{pr}_2 :\equiv \mathrm{ind}_{\sum_\alpha \beta} \left(x \mapsto \beta \left(\mathrm{pr}_1 \left(x \right) \right), a \mapsto \mathbb{1}_{\beta(a)} \right)$, i.e. $\mathrm{pr}_2((a, b)) :\equiv b$ if we are defining functions via pattern matching.

On the logical side dependent pair type can be roughly translated in to an existential claim. Let B be predicate on A, then to prove $\exists a : A, B(a)$, one need to exhibit some $a : A$ then prove that $B(a)$; this corresponds to construct a term $a : \alpha$ such that $\beta(a)$ has terms, equivalently via the constructor $(\cdot, \cdot)$ and the recursor and inductor to construct a term $(a, b) : \sum_\alpha \beta$.

A.4 Coproduct, Disjunction

The sum (or coproduct) of α and β written as $\alpha + \beta$ has two constructors $\mathrm{inl}_{\alpha+\beta} : \alpha \to \alpha + \beta$ and $\mathrm{inr}_{\alpha+\beta} : \beta \to \alpha + \beta$. The recursor states that to construct a function of type $f : \alpha + \beta \to \gamma$, it suffices to have a term $g_l : \alpha \to \gamma$ and $g_r : \beta \to \gamma$, then $f(\mathrm{inl}_{\alpha+\beta}(a)) \equiv g_l(a)$ and $f(\mathrm{inr}_{\alpha+\beta}(b)) \equiv g_r(b)$. More formally we have:

$$\mathrm{rec}_{\alpha+\beta} : \prod_{\gamma:\mathcal{U}} (\alpha \to \gamma) \to ((\beta \to \gamma) \to (\alpha + \beta \to \gamma))$$

such that $\mathrm{rec}_{\alpha+\beta}(\gamma, g_l, g_r, \mathrm{inl}_{\alpha+\beta}(a)) \equiv g_l(a)$ and $\mathrm{rec}_{\alpha+\beta}(\gamma, g_l, g_r, \mathrm{inr}_{\alpha+\beta}(b)) \equiv g_r(a)$. Similarly, the inductor which is just a dependent version of the recursor states the following:

$$\mathrm{ind}_{\alpha+\beta} : \prod_{\gamma:\alpha+\beta\to\mathcal{U}} \left(\prod_{a:\alpha} \gamma(\mathrm{inl}_{\alpha+\beta}(a))\right) \to \left(\left(\prod_{b:\beta} \gamma(\mathrm{inr}_{\alpha+\beta}(b))\right) \to \left(\prod_{x:\alpha+\beta} \gamma(x)\right)\right)$$

such that $\mathrm{ind}_{\alpha+\beta}(\gamma, g_l, g_r, \mathrm{inl}_{\alpha+\beta}(a)) \equiv g_l(a)$ and $\mathrm{ind}_{\alpha+\beta}(\gamma, g_l, g_r, \mathrm{inr}_{\alpha+\beta}(b)) \equiv g_r(b)$. Using the eliminator rules, one is justified to define functions by pattern matching $f(\mathrm{inl}_{\alpha+\beta}(a)) :\equiv \cdots$ and $f(\mathrm{inr}_{\alpha+\beta}(b)) :\equiv \cdots$ would be sufficiently determine a term $f : \alpha + \beta \to \cdots$.

On the logical side, the sum type corresponds to forming disjunction. To prove a disjunction, it suffices to prove either disjunct; this is the disjunction introduction rule. The constructors of sum type behave similarly — to construct a term of type $\alpha + \beta$, either a term of α or a term of β would suffice via the constructor $\mathrm{inl}_{\alpha+\beta}$ and $\mathrm{inr}_{\alpha+\beta}$ respectively. To prove another proposition from a disjunction, one needs to prove that the proposition follows from each disjunct alone; this is the disjunction elimination rule. The recursor and inductor behave exactly like this — to construct something from a term of sum type, one need to have "recipes" for how to construct that thing from any $a : \alpha$ and from any $b : \beta$.

References

[1] Steve Awodey. Structuralism, invariance, and univalence. *Philosophia Mathematica*, 22(1):1–11, 2014.

[2] Douglas Bridges and Erik Palmgren. Constructive Mathematics. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Summer 2018 edition, 2018.

[3] Thierry Coquand. Type Theory. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2018 edition, 2018.

[4] James Ladyman and Stuart Presnell. Identity in homotopy type theory, part i: The justification of path induction. *Philosophia Mathematica*, 23(3):386–406, 2015.

[5] James Ladyman and Stuart Presnell. Identity in homotopy type theory: Part ii, the conceptual and philosophical status of identity in hott. *Philosophia Mathematica*, 25(2):210–245, 2017.

[6] James Ladyman and Stuart Presnell. Does homotopy type theory provide a foundation for mathematics? *The British Journal for the Philosophy of Science*, 69(2):377–420, 2018.

[7] James Ladyman and Stuart Presnell. Universes and univalence in homotopy type theory. *Rev. Symb. Log.*, 12(3):426–455, 2019.

[8] Joan Moschovakis. Intuitionistic Logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2018 edition, 2018.

[9] nLab authors. empty type. `http://ncatlab.org/nlab/show/empty%20type`, May 2021. Revision 9.

[10] nLab authors. function type. `http://ncatlab.org/nlab/show/function%20type`, May 2021. Revision 13.

[11] nLab authors. product type. `http://ncatlab.org/nlab/show/product%20type`, May 2021. Revision 11.

[12] Stewart Shapiro. Identity, indiscernibility, and ante rem structuralism: The tale of i and- i. *Philosophia mathematica*, 16(3):285–309, 2008.

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

[14] Vladimir Voevodsky, Benedikt Ahrens, Daniel Grayson, et al. UniMath — a computer-checked library of univalent mathematics. available at `https://github.com/UniMath/UniMath`.