OPEN ACCESS

University of BRISTOL

Ladyman, J., & Presnell, S. (2017). Identity in homotopy type theory: Part II, the conceptual and philosophical status of identity in HoTT. *Philosophia Mathematica*, 25(2), 210-245. https://doi.org/10.1093/philmat/nkw023

Peer reviewed version

Link to published version (if available):
10.1093/philmat/nkw023

Link to publication record in Explore Bristol Research
PDF-document

University of Bristol - Explore Bristol Research

General rights

Identity in Homotopy Type Theory: Part II The Conceptual and Philosophical Status of Identity in HoTT

James Ladyman and Stuart Presnell
University of Bristol

Thursday 28th January, 2016

Abstract

One of the main motivations for Homotopy Type Theory (HoTT) is the way it treats of identity and the rich ∞-groupoid structure of types and identifications to which this gives rise. This paper investigates the conceptual and philosophical status of identity in HoTT. We examine the formal and technical features of identity types in the theory, and how these relate to other features of the theory such as its intensionality, constructive logic, and the interpretation of types as propositions and concepts. We explore the possibility that identity types might be better understood as encoding the indiscernibility of two tokens. We argue that identity types are a primitive component of HoTT.

Contents

1 Introduction

One of the main motivations for Homotopy Type Theory (HoTT) is the way it treats of identity and the rich ∞-groupoid structure of types and identifications to which this gives rise. Indeed, the homotopy interpretation of Martin-Löf's constructive intensional type theory that gives HoTT its name depends essentially upon identity types, because it is based on an interpretation of identities as paths, and between

identity types and homotopy types.[1] Furthermore, the way identity and equivalence are related by the Univalence Axiom is taken by Steve Awodey [?] to be fundamental to HoTT's claim to provide a new foundation for mathematics and to express mathematical structuralism.

As is standard in mathematics and logic, in HoTT identity is treated as a relation. In HoTT relations are types and in particular they are predicates. Predicates are functions that map tokens of one or more types to the universe $\mathcal{U}$ (for more on universes see Section **??**.[2] In particular, identity is a function type that maps pairs of tokens of a given type to $\mathcal{U}$.[3]

This paper investigates the conceptual and philosophical status of identity in HoTT. We examine the formal and technical features of identity types in the theory in Section **??**, and how these relate to other features of the theory such as its intensionality, constructive logic and the interpretation of types as propositions and concepts in Section **??**. These are matters that are not made explicit in extant accounts which focus on the mathematics.

In category theory it is said "never mistake an equivalence for an equality" [?, 33]. The axiom of Univalence, which is considered to be one of the main discoveries of the Homotopy Type Theory project, roughly says that identity is equivalent to equivalence (where equivalence is akin to isomorphism) and so embodies the idea that reasoning in mathematics should be invariant under sameness of structure. In this and other respects, the way identity is treated in HoTT departs considerably from orthodox views of identity in philosophy and logic. Indeed, these departures are so radical that they lead us to question whether 'identity' in HoTT is worthy of the name. Perhaps identity types should be interpreted instead as representing some other relation (closely related to identity). This approach offers a way of resolving many of the puzzles and problems with 'identity', and in particular

---

[1]We use the term 'HoTT' to refer to theories like that of the HoTT book that are constructive and intensional. There are theories associated with the HoTT programme that introduce the Law of the Excluded Middle and various other innovations so not everything we say about 'HoTT' should be taken as applying to all theories that might be so-called.

[2]Terminology varies in the literature – in the HoTT Book the words 'term', 'object', 'element', and 'point' are used interchangeably for what we are calling a 'token' of a type. We prefer the word 'token' for reasons we give in [?]; briefly 'term', which is the most common word used, has the connotation of something syntactic.

[3]In general, the type corresponding to the proposition that some token of some type has some property, is not the same type as that corresponding to the proposition that some other token of the type has some property. Similarly with identity types, the identity type of any pair of tokens of some type is distinct from the identity type of any other pair of tokens of that type, see below.

might make univalence seem very natural as we explain in Section **??**

In the standard presentation of HoTT, identity types are introduced as primitive components of the system. Of course, in some logics, identity is defined in terms of other resources, for example by the Hilbert-Bernays method. In Section **??** we explore the possibility that identity types might be better understood as encoding the *indiscernibility* of two tokens. This is of course closely related to the status of the Principle of the Identity of Indiscernibles (PII). However, after examining this option (and considering how discernibility and indiscernibility might be formulated, we find in Section **??** that treating identity as indiscernibility does not work. Hence, this paper provides a justification for why identity types are introduced as a primitive component of HoTT. Section **??** briefly concludes and we consider different formulations of PII in a constructive setting in the Appendix.

2   Identity in HoTT

As mentioned above, identity in HoTT is represented by identity types.[4] In accordance with the Curry-Howard correspondence, the identity type $\mathtt{Id_A(x,y)}$ expresses the proposition that tokens $\mathtt{x}$ and $\mathtt{y}$ of type $\mathtt{A}$ are identical. Note that there isn't a single identity type expressing all the identity facts about $\mathtt{A}$, but rather an identity type for each ordered pair of tokens of $\mathtt{A}$. Tokens of an identity type $\mathtt{Id_A(x,y)}$ are referred to as 'identifications' of $\mathtt{x}$ and $\mathtt{y}$. In the special case $\mathtt{Id_A(x,x)}$ we are always guaranteed to have a token of this type, since identity is reflexive. Thus the token constructor for identity types produces a token $\mathtt{refl_x : Id_A(x,x)}$ for any token $\mathtt{x : A}$. The following subsections review the most important features of identity in HoTT.

2.1   Internal versus External Identity

Fundamental to identity in HoTT is the distinction between 'external' and 'internal' identity, also described as the contrast between 'judgmental equality' and 'propositional identity'.

Given any two expressions exp_1 and exp_2 naming tokens or types, the external identity $exp_1 \equiv exp_2$ says that exp_1 and exp_2 name the same token or type, and thus are intersubstitutable in any circumstance without changing meaning.

This is a different notion from the 'internal' or 'propositional' identity, which is represented by the identity types described above. In

---

[4][**?**] explains identity types in HoTT in more detail.

particular, internal identity, being represented by a type, can be combined with other types to make more complex propositions. Judgmental equality, being external, cannot be combined in this way.

Note that clearly external identity implies internal identity. This is because, as noted above, we always have a token of $\mathtt{Id_A(x,x)}$ for any token $\mathtt{x}$, and so it follows that if we also have $\mathtt{x} \equiv \mathtt{y}$ then we get a token of $\mathtt{Id_A(x,y)}$ by substitution. If the converse implication does not hold we call the theory 'intensional' following Martin-Löf.

The fact that standard HoTT is intensional is very important as discussed in Section **??**. To avoid confusion we refer to externally non-identical tokens or types as 'distinct'.

2.2 Multiple Identifications and Higher Identities

The token constructor `refl` is the only way, without further premises, to produce an identification.

However, the intensional nature of the type theory means that the existence of some other identification of a token with itself cannot be ruled out even though there is no particular token-constructor that could produce such an additional identification. The same goes for identifications of any given pair of tokens of some type.

It is fundamental to HoTT that there can potentially be multiple identifications in any particular identity type. The homotopy interpretation, which gives the theory so much of its interest, is based on thinking of tokens of types as points in spaces, and identifications between tokens as paths. Without multiple identifications the structure of identity types would be trivial, and so there could be no interesting connection between type theory and homotopy theory. Below we show that the intensional nature of HoTT is a necessary condition for multiple identifications.

Note that in some contexts the idea of multiple identifications is quite natural especially if one thinks of identifications as like proofs of identity, since there is often more than one way to prove something. Extending this analogy, there may be proofs which seem different but are in fact identical themselves. Similarly, in HoTT for any two identifications α and β of tokens $\mathtt{a}$ and $\mathtt{b}$ in $\mathtt{A}$ there is the proposition that asserts that these two identifications are themselves identical, and this is represented by the so-called 'higher identity type' $\mathtt{Id_{Id_A(x,y)}}(\alpha, \beta)$.

This ramifies, so there is no limit in principle to the hierarchy of higher identity types; in other words, identifications of identifications may themselves be identified, and so on. It is this higher identity structure of multiple identifications at every level that gives rise to

the ∞ groupoid structure explained in the next subsection.

2.3   The Groupoid Structure of Identity

Tokens of some type and the identifications between them can be taken to be the objects and arrows of a category: `refl` gives the identity arrows, and transitivity of identity gives the associative composition of arrows.

A **groupoid** is defined to be a category in which every arrow has an inverse, and in the present case these inverses arise from the symmetry of identity. Thus types can be understood as groupoids.

Moreover, the structure of the higher identity types means that every type is an ∞-groupoid. It was known before HoTT that there is a correspondence between ∞-groupoids and homotopy theory, and the connection with constructive intensional type theory is what gave birth to Homotopy Type theory.

2.4   (Based) Path Induction

To use identifications we need the elimination rule for the identity type, called 'based path induction'. This says that for any type `A`, any token `a : A`, and any predicate `K` that can be asserted of pairs $(\mathtt{x}, \mathtt{q})$ (where `x : A` and $\mathtt{q} : \mathtt{Id_A(a, x)}$) there is a function of type

$$\mathtt{K(a, refl_a)} \to \prod_{\mathtt{(b,p):E(a)}} \mathtt{K(b, p)}$$

(Path induction is the formulation that does not involve fixing `a`.)

Path induction so stated is far from a self-evident principle governing identity. Ladyman and Presnell **?** discuss its epistemological and methodological status. Here were note that it is an essential feature of identity in HoTT that can be explained and justified either with or without the homotopy interpretation. In the next section we will make use of it to show how the intensionality of HoTT is connected to how it treats of identity.

It may seem that path induction says that all identifications are trivial. However, to prove this we would need a predicate `Q` that says of any given identification $\mathtt{p} : \mathtt{Id_A(a, b)}$ that it is identical to the trivial self-identification of some token of `A`.

$$\mathtt{Q(a, b, p)} :\equiv \mathtt{Id_{Id_A(a,a)}(p, refl_a)}$$

But such a predicate is not well-typed, because `p` is not a token of $\mathtt{Id_A(a, a)}$.

5

In the next section, we reflect on the above formal features of identity in HoTT.[5]

3   The Conceptual and Philosophical Status of Identity in HoTT

Identity types and their structure can be regarded as formal features of HoTT that are of mathematical interest independently of their conceptual and philosophical status. However, the latter is also novel and interesting and below we explicate some of its features.

3.1   Absolute or Relative?

In HoTT identity is absolute, in the sense that the identity of tokens does not vary across contexts or aspects, and there is no question of two tokens being identical in one way and not another.

In many systems we can always ask if two elements are identical to each other. For example, in ZFC, we can always ask whether a set is equal to another, whether an element of a set is equal to some set, and so on.

However, in HoTT two tokens of distinct types can never be identical and it makes no sense to ask if they're identical. The only way to ask whether `a` and `b` are identical is to ask whether an identity type is inhabited. But if `a` and `b` are of different types then there is no such identity type.

3.2   Identity and Distinctness

Identity is always of tokens of some particular type, and identity types are indexed by other types. Moreover, if `a` and `b` are of different types then we also can't assert that they are non-identical. To this extent, then, identity and non-identity are relativised to types.

Since the type of natural numbers is not identical to the type of real numbers, there is no question as to whether the natural number 2 is identical to the real number 2. (This is in roughly in accordance with how identity is treated in Michael Resnik's version of mathematical structuralism.)

This is one respect in which the way identity is treated in HoTT is contrary to orthodoxy. However, the more radical respect in which

---

[5]The constructive logic of HoTT also gives rise to nonstandard features such as non-decidable identity statements in the sense that it is not true that for all x and y either $x = y$ or $x \neq y$.

identity in HoTT is unorthodox is that, as explained in the previous section, there can exist multiple certificates to the identity type of two tokens, and higher identities between such identifications. In the next subsection we explain how the intensionality of the theory is closely related to the way it treats of identity.

3.3   Intensionality and Internal Identity

Identity, extensionality and intensionality are closely related. The identity criterion for sets expresses the extensionality of set theory. Intensionality means that substitution of co-referring terms does not always preserve truth where the meaning or mode of presentation of terms affects the truth-value of the relevant proposition. Recall that external identity entails internal identity. That the converse implication, from propositional/internal identity to judgmental equality/external identity, does *not* is definitive of an 'intensional' type theory rather than an 'extensional' one, since without external identity, two tokens cannot always be substituted for each other while preserving truth.

If the implication from internal to external identity (called 'reflection of identity') were to hold then the predicate Q defined in the last subsection of the previous section would be well typed (since from $p : Id_A(a, b)$ we could derive $a \equiv b$). Hence, we could use path induction to prove that all identities are trivial. If the latter were so the ∞-groupoid structure of higher identity types and so the connection with homotopy theory would be lost. Hence, the intensionality of HoTT is essential to it.

3.4   Types as propositions and types as concepts

As noted in Section **??**, by the Curry-Howard correspondence the identity type $Id_A(x, y)$ corresponds to the proposition that x and y are identical tokens of A. If we interpret types as propositions and tokens as certificates to propositions, then $Id_A(x, y)$ says that x and y are identical as certificates of the proposition A. This accords with the idea that we may have many ways to prove a given proposition, but some of these proofs may differ only in a trivial way and so should be considered identical.

We may alternatively think of types as concepts, with tokens as specific instances of them. (Recall that internally to the language of a constructive theory we can only say there are mathematical structures of a given type if we can actually construct a particular instance.) For

example, the type of *metric spaces* has as tokens particular metric spaces such as the Euclidean Plane.

This makes it natural for it to be possible to have identifications between distinct instances, since they may be identical qua that type, even though the symbols representing them are not externally identical. For example, consider two distinct Euclidean planes in Euclidean three-space; these are two distinct tokens of the type of *metric spaces*, but arguably we should think that they are nonetheless identical qua metric space. The intensionality of HoTT means that distinct tokens can capture the fact that we are thinking about our two copies of the Euclidean plane differently, even though thought of qua metric space we may identify them. Similarly, two tokens in one type may be identical while their counterparts in another type may not be. For example, the Hyperbolic Disc and the Euclidean Plane are not the same qua metric space but they are qua topological space. We can think about the same topological space differently by thinking about different metric spaces and forgetting about their metric structure.

Identity in HoTT is always relativized to types in this way. That is, $\mathtt{Id}_\mathtt{A}(\mathtt{x},\mathtt{y})$ does not express absolute identity between $\mathtt{x}$ and $\mathtt{y}$, but rather that $\mathtt{x}$ and $\mathtt{y}$ are identical *qua* token of type $\mathtt{A}$.

3.5   Is 'Identity' in HoTT identity?

The radical innovations in the way identity is treated in HoTT make it worth questioning whether 'identity' is the right name for the type $\mathtt{Id}_\mathtt{A}(\mathtt{x},\mathtt{y})$. Maybe, what is called 'propositional/internal identity' in HoTT is really some other kind of relation. Indeed it is often also called 'equality', and furthermore much of the work on HoTT concerns what follows when a further principle about identity called the 'Univalence Axiom' (UA) is added to the theory which relates identity and a weaker relation called 'equivalence'. In the next section, we briefly explain UA and consider its significance for our understanding of identity in HoTT. In particular, we relate UA to indiscernibility.

4   Identity, Equivalence and Univalence

Up until now we have considered the identity relation between tokens of a given type. However, we also need to consider identity between types themselves. The identity relation between types is not an addition to that between tokens as described above, because types may themselves be regarded as tokens of a higher-order type (or 'universe'). In other words, any two types $\mathtt{A}$ and $\mathtt{B}$ are tokens of some universe $\mathtt{U}$

(i.e. `A : U` and `B : U`), and thus we may form the identity type $\mathtt{Id_U(A,B)}$ as we would for any tokens of any type.

We can use the resources of the theory to define other relations between arbitrary types that we can't define between arbitrary tokens, and then consider how these relations interact with and relate to identity. A relation between types that is of particular importance is called 'equivalence' ($\simeq$).

4.1   Equivalence

Equivalence in HoTT is a relation between types that is similar to the familiar relation of isomorphism. An isomorphism between types `A` and `B` is a function $\mathtt{f : A \to B}$ having a function $\mathtt{g : B \to A}$ that is both a pre- and post-inverse:[6]

$$(\mathtt{f} \circ \mathtt{g})(\mathtt{b}) = \mathtt{b} \qquad \text{for any } \mathtt{b : B}$$
$$(\mathtt{g} \circ \mathtt{f})(\mathtt{a}) = \mathtt{a} \qquad \text{for any } \mathtt{a : A}$$

Rather than requiring that a single function `g` serves as both pre- and post-inverse, we could ask instead for functions $\mathtt{h_1, h_2 : B \to A}$ such that $\mathtt{h_1}$ is a pre-inverse of `f` and $\mathtt{h_2}$ is a post-inverse of `f`:

$$(\mathtt{f} \circ \mathtt{h_1})(\mathtt{b}) = \mathtt{b} \qquad \text{for any } \mathtt{b : B}$$
$$(\mathtt{h_2} \circ \mathtt{f})(\mathtt{a}) = \mathtt{a} \qquad \text{for any } \mathtt{a : A}$$

The type of equivalences between types `A` and `B`, which we write as $\mathtt{Equiv(A,B)}$, has as its tokens triples $(\mathtt{f}, \mathtt{h_1}, \mathtt{h_2})$ where $\mathtt{h_1}$ and $\mathtt{h_2}$ are respectively pre-and post-inverses to `f`. Every function that is an isomorphism is also an equivalence, and it is a theorem [**?**, Section 2.4] that from any equivalence we can produce an isomorphism.

4.2   Univalence

The Univalence Axiom (UA) relates identity between types to equivalence. UA says, roughly, that identity and equivalence are equivalent (as mnemonic we can write $=\simeq\simeq$ but this makes no formal sense). More precisely it says that types of the form $\mathtt{Id_U(A,B)}$ are equivalent to the corresponding types of the form $\mathtt{Equiv(A,B)}$, i.e.

$$\mathtt{UA : Equiv\big(Id_U(A,B),\ Equiv(A,B)\big)}$$

Note that UA can be applied to the equivalence between identity types and equivalence types. That is, from the equivalence between

---

[6] Pre- and post-inverse are more commonly called right- and left-inverses.

$\text{Id}_\text{U}(\text{A},\text{B})$ and $\text{Equiv}(\text{A},\text{B})$ asserted by UA we can derive (by an application of UA) an identification of $\text{Id}_\text{U}(\text{A},\text{B})$ and $\text{Equiv}(\text{A},\text{B})$.

It might appear that the equivalence (and therefore identity) between the identity and equivalence relations means that one or other of these relations can be dispensed with. Perhaps UA means that identity does not need to be introduced as a primitive of the language, since it can be replaced with equivalence. This is not the case. Recall that the equivalence relation is only defined between types, not between tokens of arbitrary types. Thus UA says only that identity and equivalence are equivalent *where both are defined*. It does not allow us to eliminate identity from the language, since UA is a statement about equivalence and identity between types in a universe (not identity within any particular types) so we still need a way to express identity between tokens of arbitrary types. (Furthermore there is no external analogue of equivalence so we still need external identity.)

4.3 Univalence and Intensionality

Univalence is sometimes said to be a kind of extensionality principle because in HoTT without UA the product $\text{A} \times \text{B}$ cannot be proved to be identical to the product $\text{B} \times \text{A}$ but with univalence it can. Furthermore, $UA \Rightarrow$ Function Extensionality, which identifies functions that have the same input-output behaviour [**?**]. Note that $= \not\Rightarrow \equiv$ still holds so HoTT with UA is still intensional in the sense defined above.

4.4 Justifying UA

The Homotopy Type Theory research programme is closely associated with the idea of 'Univalent Foundations' due to Voeveodesky, and UA is taken to be one of the most important new discoveries arising from this approach to mathematics. It is said by Awodey and Michael Shulman that UA gives rise to a 'different understanding of identity'.

However, as noted above, even with univalence we cannot replace identity by equivalence, and the two notions are not externally equal (i.e. $= \not\equiv \simeq$).

Given that the difference between equivalence and identity is still recognised in standard HoTT, what justifies their unification by UA?[7]

Awodey claims that UA is equivalent to the Principle of Invariance (PI) which states that all reasoning internal to the theory is invariant under isomorphism ([**?**]). He relates univalence to mathematical

---

[7]There are approaches broadly within the HoTT research programme in which equivalence and identity are the same by definition

structuralism, which he expresses in terms of the idea that isomorphic structures are the same. If equivalence rather than isomorphism amounts to sameness of structure then arguably structuralism motivates the principle that reasoning should be invariant under equivalence. Elsewhere we argue that this is not sufficient to justify UA [**?**]. Another approach to justifying it begins from the observation made in Section **??** that in HoTT identity is always *identity of tokens in a type*. This suggests the idea of identity qua some structure – for example, the Euclidean plane and the Hyperbolic disc are not identical *simpliciter*, but they do correspond to tokens of the type *qua* topological space that are identical in that type. Relatedly, the things we can say within the language of HoTT about some entity depend upon what type that entity belongs to, i.e. as what kind of structure it is being regarded.

Different mathematical theories are designed to have the resources to describe mathematical structures of different types, forgetting about other features. For example, the permutation group of three objects and the symmetry group of the triangle are identical qua groups so we can regard them as distinct tokens between which there are identities. If we take 'equivalent' to mean indiscernible within some type then IP follows since reasoning about types can only use the properties defined for the type in question. This suggests that in full HoTT including identity types we can read Id as 'indiscernibility'. On this reading of the Id relation between types, it is natural to take indiscernibility to be equivalent to equivalence. Perhaps also if we took $\mathtt{Id_A(a,b)}$ to denote indiscernibility rather than identity, any problems with thinking of identity types as expressing identity would dissipate and the motivation for principles such as Univalence would become clearer.

If we define a relation of indiscernibility within the language we can explore its relation to Id (in the same way that we can define a relation on $\mathbb{N}$ that's provably equivalent to $\mathtt{Id_{\mathbb{N}}}$, and likewise for coproducts). This brings us to the Principle of the Identity of Indiscernibles (PII) and the question as to whether identity needs to be taken as primitive in HoTT or whether it might be reducible. If the interpretation of Id as indiscernibility works, might we go further and *replace* Id with $\mathtt{InDis}$ (or some similar relation). In the next section we consider discernibility, indiscernibility and PII in HoTT.

11

5 Identity, Discernibility and Indiscernibility and PII in HoTT

It is easy enough to represent discernibility and indiscernibility in HoTT but the correct formulation of PII in the context of HoTT is a very complicated issue, largely because the constructive logic of the theory means that classically equivalent statements may mean very different things. In particular, as shown below, in HoTT natural interpretations of 'indiscernible' and 'not discernible' do not mean the same thing, because the latter just means we do not have a predicate that applies to one and not the other, whereas the former expresses the stronger claim that we have a way of deducing that a predicate applies to both if to one of two tokens of a type. Furthermore standard statements of PII assume the classical equivalences between conditionals and disjunctions and conjunctions that do not hold constructively. This means that PII comes in many different from. The formal definitions in HoTT of discernibility and indiscernibility adopted in what follows are given in the next two subsections.

5.1 Discernibility

A natural definition of discernibility is

$$\text{Dis}_{\text{A}}(\text{a}, \text{b}) :\equiv \sum_{\text{P}:\text{A}\rightarrow\mathcal{U}} \text{P}(\text{a}) \times \neg\text{P}(\text{b})$$

which asserts that there is some property – a 'distinguisher' – that holds of a and not of b.

Note that despite the asymmetry in the statement of the definition this relation is symmetric, since if we have $(\text{Q}, \text{q}_{\text{a}}, \bar{\text{q}}_{\text{b}}) : \text{Dis}_{\text{A}}(\text{a}, \text{b})$ then by Double-Negation Introduction we have $(\neg\text{Q}, \bar{\text{q}}_{\text{b}}, \bar{\bar{\text{q}}}_{\text{a}}) : \text{Dis}_{\text{A}}(\text{b}, \text{a})$.

Note that the requirement that there be a one-place predicate that applies to one object and not the other is equivalent to *absolute discernibility* in standard logic (assuming that for every formula free in one variable there is a corresponding monadic predicate). While much recent debate about PII has involved the notion of *weak discernibility*, this notion can be set aside in HoTT, since weak discernibility collapses to absolute discernibility. This is because predicates are function types, and a function type free in two types can always be used to make function types free in one type by currying. (Note that in general a predicate being one-placed says nothing about the metaphysics of the corresponding property, as with the monadic predicate 'is an uncle' and the relational not intrinsic property of being an uncle.)

12

5.2 Indiscernibility

A natural definition of indiscernibility is

$$\text{InDis}_A(a, b) :\equiv \prod_{P:A \to \mathcal{U}} P(a) \leftrightarrow P(b)$$

where the notation '$X \leftrightarrow Y$' is an abbreviation for $(X \to Y) \times (Y \to X)$.

Note that this form mirrors exactly standard statements of indiscernibility in the literature on PII quantifying over all predicates and saying that any that applies to one applies to the other and vice versa.

Clearly $\text{InDis}_A(a, b)$ is symmetric, reflexive, and transitive, and supports substitution: if we have $P(a)$ (for some property P) and $\text{InDis}_A(a, b)$ then we have $P(b)$. It is therefore of interest to see what parallels can be drawn between the behaviour of $\text{Id}_A(a, b)$ and of $\text{InDis}_A(a, b)$ in HoTT.

Perhaps, as suggested above, the unusual features of the treatment of identity in HoTT are collectively telling us that 'identity' types are simply mis-named, and that the relation they represent is not identity at all but rather *indiscernibility*. Rather than supplementing HoTT$^-$ with the addition of identity types, as we do in HoTT, we would instead define InDis within HoTT$^-$ show (if possible) that it satisfies the defining properties of the identity type in HoTT, and then use this in HoTT$^-$ in place of identity types.

We have the same type construction (given a type and two tokens we can form the corresponding indiscernibility type). For any $a : A$ we have a token $\text{refl}'_a : \text{InDis}_A(a, a)$ given by the function that returns for any P the identity function on $P(a)$. As mentioned above Indiscernibility is an equivalence relation. We could introduce path induction by stipulation, by defining $E'(a) :\equiv \sum_{x:A} \text{InDis}_A(a, x)$ and asserting that for any token $(b, p) : E'(a)$ we have $\text{InDis}\big((b, p), (a, \text{refl}'_a)\big)$.

5.3 To Discern or not and to Indiscern or not

What is the relationship between InDis, Dis, and their (single and double) negations? Clearly InDis and Dis are of opposite 'valence', but what entailments hold between them?

First, it is clear that InDis and Dis are contrary to one another:

$$(\text{InDis} \times \text{Dis}) \to 0$$

Thus each entails the negation of the other:

$$\text{InDis} \vdash \neg \text{Dis}$$
$$\text{Dis} \vdash \neg \text{InDis}$$

and so by contraposition (and Double Negation Introduction) we have

$$\texttt{Dis} \vdash \neg\neg\texttt{Dis} \vdash \neg\texttt{InDis}$$
$$\texttt{InDis} \vdash \neg\neg\texttt{InDis} \vdash \neg\texttt{Dis}$$

By contraposition and Triple Negation Elimination,

$$\neg\texttt{InDis} \vdash \neg\neg\texttt{Dis} \quad \text{if and only if} \quad \neg\texttt{Dis} \vdash \neg\neg\texttt{InDis}$$

and these entailments follow from either of $\neg\texttt{InDis} \vdash \texttt{Dis}$ or $\neg\texttt{Dis} \vdash \texttt{InDis}$.

The negation of indiscernibility, $\neg\texttt{InDis}_{\texttt{A}}(\texttt{a}, \texttt{b})$, implies

$$\sum_{\texttt{P}:\texttt{A}\to\mathcal{U}} \neg\big(\texttt{P}(\texttt{a}) \leftrightarrow \texttt{P}(\texttt{b})\big)$$

However, from the negation of a conjunction we cannot derive the negation of either conjunct, so for a given $\texttt{P}$ we cannot in general determine from $\neg\texttt{InDis}_{\texttt{A}}(\texttt{a}, \texttt{b})$ which of $\texttt{P}(\texttt{a}) \to \texttt{P}(\texttt{b})$ or $\texttt{P}(\texttt{b}) \to \texttt{P}(\texttt{a})$ fails. Thus, for many applications, $\neg\texttt{InDis}_{\texttt{A}}(\texttt{a}, \texttt{b})$ may be too weak to be useful and we need the positive characterisation of discernibility above.

$\neg\texttt{Dis}$ is equivalent to

$$\prod_{\texttt{P}:\texttt{A}\to\mathcal{U}} \neg\big(\texttt{P}(\texttt{a}) \times \neg\texttt{P}(\texttt{b})\big)$$

Since $\neg(\texttt{X} \times \neg\texttt{Y})$ is in general weaker than $(\texttt{X} \to \texttt{Y})$, $\neg\texttt{Dis}$ does not entail $\texttt{InDis}$.

5.4 The Indiscernibility of Identicals

An important feature of identity in HoTT is that identical tokens share all their properties: for any type $\texttt{A}$ and any predicate $\texttt{P} : \texttt{A} \to \mathcal{U}$, there is a function

$$\texttt{f} : \prod_{\texttt{x},\texttt{y}:\texttt{A}} \prod_{\iota:\texttt{Id}_{\texttt{A}}(\texttt{x},\texttt{y})} \texttt{P}(\texttt{x}) \leftrightarrow \texttt{P}(\texttt{y})$$

(where $\texttt{A} \leftrightarrow \texttt{B}$ abbreviates $(\texttt{A} \to \texttt{B}) \times (\texttt{B} \to \texttt{A})$), and in particular for any $\texttt{x} : \texttt{A}$

$$\texttt{f}(\texttt{x}, \texttt{x}, \texttt{refl}_{\texttt{x}}) :\equiv \big(\texttt{id}_{\texttt{P}(\texttt{x})}, \texttt{id}_{\texttt{P}(\texttt{x})}\big)$$

Thus identical tokens are indiscernible. (This is proved in the HoTT Book as a consequence of path induction in Section 1.12.) (Note that of course external identity therefore also implies indiscernibility.)

5.5   Indiscernibility and Language

Since the definition of indiscernibility involves quantification over predicates we must pay attention to what language we are working in (especially when we considerng using indiscernibility to extend one language to resemble another). PII is trivially true if its scope includes predicates representing haecceities since everything is discernible from everything else by such predicates.

We could just formulate `InDis` in HoTT$^-$, but to express PII we need identity types in the language as well. In a language such as HoTT containing identity types, given any $a : A$ we can construct the predicate $\mathtt{Id}_A(a)$. Then, since this predicate holds of a it must also hold of any b that is indiscernible from a. Thus in such a language indiscernibility entails identity, and so PII follows trivially.

We might try to define the notion of *non-identity-involving* (NII) predicates within HoTT. The obvious way to (try to) do this would be to define NII recursively, with some base cases and some rules for construction. If we could do this correctly, then we could restrict attention to NII predicates, and thus consider whether PII holds nontrivially in HoTT.

In the next section we consider how PII should be formulated assuming the definition of `InDis` however the quantification over predicates is restricted.

5.6   The identity of Indiscernibles

PII is usually stated as follows: For every two objects, if for any property, one has it if and only if the other has it, then they are identical. We can state this in HoTT as follows:

$$\prod_{A:\mathcal{U}} \prod_{a,b:A} \big(\mathtt{InDis}_A(a,b) \to \mathtt{Id}_A(a,b)\big)$$

The problem is that PII is just as usually stated in contrapositive form as follows: There are no two objects that share all their properties. Indeed, Leibniz's formulation of PII is "it is not true that two substances may be exactly alike and differ only numerically, solo numero". [8]

The most direct translation of this is:

$$\neg \sum_{x,y:C} \big(\mathtt{Alike}(x,y) \times \neg\mathtt{Id}_C(x,y)\big)$$

---

[8]Leibniz gives an argument from haecceities in the *Discourse on Metaphysics*. He also argues for PII from the Principle of Sufficient Reason (if objects differed solo numero then God would have no reason to arrange them one way rather than another).

where $\texttt{Alike}(\texttt{x}, \texttt{y})$ expresses that $\texttt{x}$ and $\texttt{y}$ have all properties in common, and we take "differ only numerically, solo numero" to mean simply that $\texttt{x}$ and $\texttt{y}$ are not identical.

But constructively this is a very different statement from the positive claim above because in constructive logic properties and relations do not either hold or not hold. Hence, it is very difficult to say how PII should be formulated in HoTT. There are three strengths of material implication depending on whether it is put in terms of the conditional or conjunction or disjunction. This can apply to both the principle as a whole and within it so we have potentially nine forms of PII. These are presented and briefly analysed in the Appendix. In the next section we give two arguments for the claim that, regardless of the form or status of PII in HoTT, identity types must be taken as primitive components of the theory.

6 Identity is Primitive in HoTT

Unlike identity, discernibility and indiscernibility quantify over predicates and so they should be indexed according to levels of the hierarchy of types as explained in the next subsection. We argue there that there are important differences between $\texttt{Id}$ and $\texttt{InDis}$, and that this means that identity types cannot be reconstrued as indiscernibility types and are primitive elements of the theory.[9] In the next subsections, we consider identity and indiscernibility in the context of 'finite types', and then in types in general. The penultimate subsection considers how case analysis relates to 'uniqueness principles' for types. In the final subsection we consider the status of identity and diversity facts in HoTT

6.1 Universes

In the definition of $\texttt{InDis}$ we have written the type of the predicates that we quantify over as $\texttt{A} \to \mathcal{U}$, where $\mathcal{U}$ is the universe under consideration.[10] But since no universe contains all types we cannot pick in advance a single universe to be the domain of all our mathematical reasoning. We must therefore consider the possibility of different versions of $\texttt{InDis}$ that quantify over predicates mapping into different universes. That is, rather than saying that two tokens are simply

---

[9]Whether we can plausibly retain the informal interpretation of $\texttt{Id}$ as 'indiscernibility' and use it to justify UA is not considered further here.

[10] For a more detailed discussion of universes in HoTT, including an explanation of their most important properties, see [**?**, Section 2].

'indiscernible', we should say that they are 'indiscernible relative to universe $\mathcal{U}$', 'indiscernible relative to universe $\mathcal{V}$' (i.e. they share all properties that map into that universe) etc., where for any $\mathcal{V}$,

$$\texttt{InDis}_{\texttt{A}}^{\mathcal{V}}(\texttt{a},\texttt{b}) :\equiv \prod_{\texttt{P:A}\to\mathcal{V}} \texttt{P(a)} \leftrightarrow \texttt{P(b)}$$

A failure of indiscernibility relative to $\mathcal{U}$ implies a failure of indiscernibility relative to $\mathcal{U}'$ for arbitrary $\mathcal{U} : \mathcal{U}'$, since universes are cumulative. However, it is not clear whether the fact that two tokens are indiscernible relative to $\mathcal{U}$ implies that they are also indiscernible relative to $\mathcal{V}$ for arbitrary universes $\mathcal{U}$ and $\mathcal{V}$.[11] Since there is no type or universe of all universes, we cannot quantify over them to express 'indiscernibility relative to all universes'

One response to this would be to re-interpret the statement of indiscernibility via *typical ambiguity* ([?], [?, Section 2]). That is, rather than taking '$\mathcal{U}$' to denote one particular universe, we could take it as a 'dummy variable' that can stand for any universe. Thus the intended meaning of the statement of `InDis` would be that *whatever* universe is introduced, the two tokens are indiscernible with respect to all predicates mapping into that universe.

This is a mismatch between identity types and indiscernibility types, since the former express a single relation, while the latter, according to this proposal, collectively express an indexed family of relations. PII would then be a schema and all instances of it could be true but no finite collection of them would express the reducibility of identity to indiscernibility.

6.2   Finite Types

The debate about mathematical structuralism seems to have concluded that even PII formulated in terms of weak discernibility fails to hold for some mathematical structures such as edgeless graphs because they arguably contain elements that are utterly indiscernible (see [?] and [?]). We might therefore expect a similar phenomenon to arise for finite types in HoTT, i.e. types that have a finite number of *nullary* (no input) token constructors and no others (such as the unit and binary types that are of fundamental importance). However, we show that this is not the case.

Consider, for example, the type $\texttt{2}$ with two token constructors producing tokens $\alpha : \texttt{2}$ and $\beta : \texttt{2}$. To discern α and β we need a predicate of type $\texttt{2} \to \mathcal{U}$ that holds of one and not the other. Recall

---

[11] It may be the case that a theorem of this kind holds, but it has not yet been proved as far as we know.

that predicates on a given type, like all functions, are given by the recursion and induction rules for that type. In the present case, the recursion rule for 2 [?, Section 1.8] says that to define a predicate $P : 2 \to \mathcal{U}$ we must pick out two types $P(\alpha)$ and $P(\beta)$, and any choice of two types gives a predicate on 2. In other words, a predicate on 2 is completely specified in terms of what it does with tokens α and β.

In particular, let P be the predicate that maps α to the Unit type 1 and β to the Zero type 0, and vice versa for Q. Clearly we have a token of $P(\alpha) \times \neg P(\beta)$ and a token of $Q(\beta) \times \neg Q(\alpha)$. These predicates therefore discern α and β.

More generally, a predicate on any finite type is defined by the recursion rule for that type, which says that to define a predicate we must pick out a type for each constructor of the finite type, and that any such choice defines a predicate. Thus for any finite type F and for each constructor producing a token $f : F$ we can define a 'characteristic predicate' like P and Q above that returns 1 when given f and returns 0 when given any other token of F.

Thus by use of characteristic predicates all externally distinct tokens of finite types can be discerned – for tokens x, y of any finite type, if $x \not\equiv y$ then $\text{Dis}(x, y)$ (and of course, if $x \equiv y$ then $\text{InDis}(x, y)$, trivially). In this sense the rules for defining functions on finite types reflect the facts about the external identity and distinctness of tokens of those types into theorems about their indiscernibility and discernibility (respectively).

One might object that this discernment is somehow illicitly using identity because, for example, the predicates P and Q need to be able to tell the difference between α and β in order to give the right output when given a particular input. However, recall that predicates, like all functions, are defined via substitution on expressions. The predicates P and Q are not given as their inputs the tokens α and β but rather *expressions denoting* α and β. Hence, this kind of discernment is via *names*. In HoTT every token of every type has a name which is an expression that refers uniquely to that token and to no other. Of course, the name of a token is arbitrary, and could be changed to any other expression (as long as this is done consistently and without collision with any expression already in use). We are assuming that the distinctness and discernibility of expressions is not uncertain, since this is a basic requirement of being able to use expressions at all. Thus the definition of P and Q does depend upon a pre-existing notion of identity, but this is the *external* judgmental equality relation $\equiv$ that holds between expressions, rather than the internal propositional identity represented by identity types.

6.3   Generalising from Finite Types

We turn now from finite types to types in general. For an arbitrary
type X it is still the case that predicates on X are defined via the
recursion/induction rules. In some cases we will be able to define
characteristic predicates for each token. For example, for the natural
numbers $\mathbb{N}$ (with zero element $0_{\mathbb{N}} : \mathbb{N}$ and successor function $\mathtt{s} : \mathbb{N} \to$
$\mathbb{N}$) we can define the following family of predicates $\chi_{\mathtt{i}} : \mathbb{N} \to \mathcal{U}$ (for
$\mathtt{i} : \mathbb{N}$)

$$\chi_{0_{\mathbb{N}}}(0_{\mathbb{N}}) :\equiv \mathtt{1}$$
$$\chi_{0_{\mathbb{N}}}(\mathtt{s(m)}) :\equiv \mathtt{0}$$
$$\chi_{\mathtt{s(n)}}(0_{\mathbb{N}}) :\equiv \mathtt{0}$$
$$\chi_{\mathtt{s(n)}}(\mathtt{s(m)}) :\equiv \chi_{\mathtt{n}}(\mathtt{m})$$

Thus for any $\mathtt{n} : \mathbb{N}$ we have $\chi_{\mathtt{n}}(\mathtt{n}) \equiv \mathtt{1}$, and $\chi_{\mathtt{n}}(\mathtt{j}) \equiv \mathtt{0}$ for all $\mathtt{j} \not\equiv \mathtt{n}$, so
$\chi_{\mathtt{n}}$ is a characteristic predicate for $\mathtt{n}$.

The recursion rule for an arbitrary type X will (in most cases) say
that in order to define a predicate (or any function) on X it suffices
to specify what that predicate does to the outputs of the constructors
for X. [**?**, Section 5.6] Thus arbitrary *constructed tokens* (i.e. tokens
arising as the output of a constructor) can in general be discerned
by characteristic predicates. However, we cannot assume that *every*
token is externally identical to the output of a constructor, and so it
does not follow that all tokens can be discerned.

This is good news – if we could define characteristic predicates for
all tokens then, as in the case of finite types, tokens would be dis-
cernible iff they were judgmentally distinct, and so (in)discernibility
would reflect external identity into the language. Thus taking in-
discernibility as a substitute for identity would give an *extensional*
theory in which no (non-trivial) 'higher identities' could exist, and so
the distinctive features of HoTT explained in Section **??** – such as
the analogy with homotopy theory and the ∞-groupoid structure of
identity types – would be lost.

Moreover it would be contradictory then to add to the language
a separate internal identity predicate (with a corresponding trans-
port function defined as in HoTT) which identified tokens that are
externally distinct, since for any $a = b$ this would give a function
$\chi_a(a) \to \chi_a(b)$, i.e. $\mathtt{1} \to \mathtt{0}$, so producing contradiction.[12]

---

[12] However, Voevodsky [**??**] is investigating ways to define a language with two internal
identity relations, one extensional and the other intensional, so evidently it is possible to
have two such relations in a language if suitable constraints are imposed.

6.4   Case analysis

An alternative way to write function (and predicate) definitions is 'case analysis' or 'pattern matching' style; for example, the definition of the predicate $P : 2 \to \mathcal{U}$ considered in Section **??** may be written as

$$P(\alpha) :\equiv 1$$
$$P(\beta) :\equiv 0$$

Such a definition might be understood as standing for an 'if . . . then . . . ' expression; specifically, either

if $x \equiv \alpha$ then $P(x) :\equiv 1$; else if $x \equiv \beta$ then $P(x) :\equiv 0$

or, if we have identity types in the language, perhaps as

if $x =_2 \alpha$ then $P(x) :\equiv 1$; else if $x =_2 \beta$ then $P(x) :\equiv 0$

One might object to such a definition on the grounds that it only says what to do with the constructed tokens, and says nothing about how to handle arbitrary tokens. In order for such a definition to be adequate we need some assurance of *exhaustivity* – either that no other tokens of the type exist, or that the value of P at any other token is fully determined by the above specification.

In HoTT this is resolved by proving a Uniqueness Principle for the type, which (in many cases) says that every token of the type is equal to some constructed token.[13] For example, for the finite type 2 described above the Uniqueness Principle says [**?**, Equation 1.8.1]

$$\prod_{x:2} \mathtt{Id}_2(x, \alpha) + \mathtt{Id}_2(x, \beta)$$

Thus given any arbitrary token x of the type, the uniqueness principle gives an identification between x and some constructed token c, and then transport along this identification gives an identification between $f(x)$ and $f(c)$, where $f(c)$ is given explicitly by the function definition.[14] Thus the uniqueness principle ensures that although the function is only explicitly given values at constructed tokens, it

---

[13] This is not always what the uniqueness principle for a type says: for example, for identity types the uniqueness principle doesn't say that every identification is equal to a trivial self-identification (which statement is not well-typed), but rather that every identification has a counterpart in a based identity type that is equal to the counterpart of a trivial self-identification. See [**?**, Section 6.2].

[14] A similar situation holds when the uniqueness principle doesn't give an identification between arbitrary tokens and constructed tokens, as discussed in footnote **??**.

is nonetheless well-defined for all tokens of the type, up to internal identity.

Since uniqueness principles directly involve identifications they of course cannot be stated in the language without identity types that we consider here, and so this assurance of exhaustivity is not available. However, wherever in HoTT we could prove a Uniqueness Principle for a type saying that every token of the type is *equal* to a constructed token, we expect to be able to prove a corresponding Indiscernibility Principle saying that every token of the type is *indiscernible* from a constructed token. For example, for the finite type 2 the Indiscernibility Principle (corresponding to the above Uniqueness Principle) says

$$\prod_{\mathtt{x}:2} \mathtt{InDis_2(x,}\alpha) + \mathtt{InDis_2(x,}\beta)$$

To prove this we use the induction principle for 2, which says that to derive a dependent function of type $\prod_{\mathtt{x}:2} \mathtt{T(x)}$ (for any predicate $\mathtt{T} : 2 \to \mathcal{U}$) it is sufficient to give its values at α and β. Since we trivially have tokens of $\mathtt{InDis_2}(\alpha,\alpha)$ and $\mathtt{InDis_2}(\beta,\beta)$ the Indiscernibility Principle for 2 follows immediately.[15]

Thus, when given an arbitrary token $\mathtt{x} : 2$ to which $\mathtt{P}$ is to be applied, we may apply the dependent function witnessing the above Indiscernibility Principle to this token, thus yielding either a token of $\mathtt{InDis_2(x,}\alpha)$ or a token of $\mathtt{InDis_2(x,}\beta)$. From the resulting token we extract either a function of type $\mathtt{P}(\alpha) \to \mathtt{P(x)}$ or a function of type $\mathtt{P}(\beta) \to \mathtt{P(x)}$. Thus the Indiscernibility Principle ensures that function definitions in the above style are indeed exhaustive.

6.5 The grounding of identity and diversity facts

We have seen that even without identity types in the language we can recover the expected facts about discernibility and indiscernibility of tokens. In particular, for any uniqueness principle in HoTT we expect in general to have a corresponding indiscernibility principle in HoTT$^-$. The proofs of these principles and the construction of characteristic predicates (where these are possible) indicates that the facts about identity and diversity of tokens are *built into* the recursion/induction rules for types. In many cases, the definition of the recursion/induction rule for a type encodes the intention that all

---

[15] This argument may appear circular: to justify the claim that function definitions via the recursion/induction rules are adequate (i.e. exhaustive) we prove a theorem by constructing a function defined via the same recursion/induction rules. We address this in the next subsection.

tokens of the type are the outputs of constructors, and the uniqueness or indiscernibility principles then reflect this intention into the language.[16] In sum, in HoTT type definitions involve both constructors and recursion/induction rules. Since the latter control what can be said in the theory about the tokens of a type, they enable us to control what tokens can be said to exist in the type (up to identity or indiscernibility in the type). Very roughly, we may say that the constructors set a lower bound on what tokens exist, while the recursion/induction rules set an upper bound on what tokens exist.

In the light of the above, it is clear that identity and diversity facts in HoTT are not grounded in qualitative facts, since the latter are all the facts that are determined by what can be predicated of what, and as we have seen, predicates are functions, and the rules for functions already encode the identity and diversity facts. In this sense, identity and diversity facts are primitive, however, they can also be seen as grounded in the recursion/induction rules.

7   Conclusion

Both the intensionality of HoTT and its constructive nature are important for the understanding of identity types. The very definition of intensionality requires that identity types do not reflect external identity, and even with UA equivalence and internal identity are not externally identical. Constructive logic considerably complicates indiscernibility in the theory and its relation to identity types. Identity types cannot be reconstrued as indiscernibility types, since however these are defined they must be indexed to Universes where identity types are not. Furthermore, we can always discern distinct constructed tokens, but in general, because of the intensionality of theory, there can be externally distinct tokens that are not internally discernible.

We raised the question of whether internal identity in HoTT is worthy of the name. Clearly, there is a different notion of external identity which lacks the novel features of identity types. Whether we can plausibly retain the informal interpretation of `Id` as 'indiscernibility' and use it to justify UA along the lines suggested above is not considered further here. Given a clear understanding of the status of identity types in HoTT and how they work what we call them is somewhat moot. They are essential in HoTT for the formulation of UA, and because equivalence does not apply to tokens that are not

---

[16] However, these principles do not rule out 'exotic models' in which (from an external point of view) a type may have additional elements that go beyond the intended content of the type [?].

themselves types. Identity types are the internal way of representing identity facts in general, and so it seems right to say that what is expressed by identity types is identity rather than some other relation, and hence that identity is indeed treated in a nonstandard way in HoTT.

8   Appendix: What's the correct formulation of PII in HoTT?

Recall that the most direct translation of Leibniz's formulation (assuming we have fixed a type C) is:

$$\neg \sum_{\text{x,y:C}} \big(\texttt{Alike}(\text{x}, \text{y}) \times \neg \texttt{Id}_{\texttt{C}}(\text{x}, \text{y}) \big)$$

where $\texttt{Alike}(\text{x}, \text{y})$ expresses that x and y have all properties in common. Below we consider different definitions of $\texttt{Alike}$. First we consider different forms of PII that arise however it is defined.[17]

Since not-exists and forall-not are constructively equivalent, we can rearrange the above to:

$$\prod_{\text{x,y:C}} \neg \big(\texttt{Alike}(\text{x}, \text{y}) \times \neg \texttt{Id}_{\texttt{A}}(\text{x}, \text{y}) \big)$$

Arranged this way, the body of the quantified proposition has the form of one of the three (classically equivalent) variants of material conditional:

$$\neg \texttt{A} + \texttt{B} \;\vdash\; \texttt{A} \to \texttt{B} \;\vdash\; \neg(\texttt{A} \times \neg \texttt{B})$$

where constructively both entailments are strictly one-way.

Note that $\neg(\texttt{A} \times \neg \texttt{B})$ is equivalent to $\texttt{A} \to \neg\neg \texttt{B}$, so we'll use the latter variant.

8.1   Variants of PII

We should therefore consider three variants of PII:

$$PII^{++} :\equiv \prod_{\text{x,y:C}} \big(\neg \texttt{Alike}(\text{x}, \text{y}) + \texttt{Id}_{\texttt{A}}(\text{x}, \text{y}) \big)$$

---

[17]Analogously different formulations of PII arise in any constructive setting making the below taxonomy applicable more widely. Of course all these formulations of PII are trivially true unless the scope of quantification over predicates is restricted somehow as explained above.

$$PII^+ :\equiv \prod_{\text{x,y:C}} \big(\texttt{Alike}(\text{x},\text{y}) \to \texttt{Id}_\texttt{A}(\text{x},\text{y})\big)$$

$$PII^- :\equiv \prod_{\text{x,y:C}} \big(\texttt{Alike}(\text{x},\text{y}) \to \neg\neg\texttt{Id}_\texttt{A}(\text{x},\text{y})\big)$$

PII^{++} seems too strong to be reasonable: it says that for any two tokens of C we know either that they are identical or that they are not alike. But constructively we can be in the situation where there are two contrary possibilities but we know neither of them. So we would not expect PII^{++} to hold for most types, and we won't consider it further for now.

For a given $\text{x}, \text{y} : \text{C}$, the type $\texttt{Alike}(\text{x},\text{y})$ says that x and y share all their properties. Thus it involves a quantification over predicates on C. As with PII this has multiple formulations corresponding to the different variants of the material conditional.

8.1.1 Variants of Alike

$$\texttt{Alike}^{++}(\text{x},\text{y}) :\equiv \prod_{\text{P:A}\to\mathcal{U}} \big(\neg\texttt{P}(\text{x}) + \texttt{P}(\text{y})\big) \times \big(\neg\texttt{P}(\text{y}) + \texttt{P}(\text{x})\big)$$

$$\texttt{Alike}^+(\text{x},\text{y}) :\equiv \prod_{\text{P:A}\to\mathcal{U}} \texttt{P}(\text{x}) \leftrightarrow \texttt{P}(\text{y})$$

$$\texttt{Alike}^-(\text{x},\text{y}) :\equiv \prod_{\text{P:A}\to\mathcal{U}} \neg\big(\texttt{P}(\text{x}) \times \neg\texttt{P}(\text{y})\big) \times \neg\big(\texttt{P}(\text{y}) \times \neg\texttt{P}(\text{x})\big)$$

8.1.2 Elimination of the Strongest (again)

By distributivity, $\texttt{Alike}^{++}(\text{x},\text{y})$ can be reformulated as

$$\prod_{\text{P:A}\to\mathcal{U}} \big(\texttt{P}(\text{x}) \times \texttt{P}(\text{y})\big) + \big(\neg\texttt{P}(\text{x}) \times \neg\texttt{P}(\text{y})\big)$$

However, to capture the intended meaning of 'alike' we must have $\texttt{Alike}(\text{x},\text{x})$ for any $\text{x} : \text{C}$, and so $\texttt{Alike}^{++}(\text{x},\text{x}) \equiv \prod_{\text{P:A}\to\mathcal{U}} \texttt{P}(\text{x}) + \neg\texttt{P}(\text{x})$ is too strong, since it asserts that every predicate on C is decidable.

We therefore set aside $\texttt{Alike}^{++}$ and don't consider it further.

8.2   Four Forms of PII

$\mathtt{Alike}^+$ is just $\mathtt{InDis}$ as defined above.

By a constructive de Morgan law, $\mathtt{Alike}^-(\mathtt{x},\mathtt{y})$ is just the negation of $\mathtt{Dis}(\mathtt{x},\mathtt{y})$ as defined above.

Putting these variants together, we therefore get four forms of PII:

$$PII_+^+ :\equiv \mathtt{Alike}^+(\mathtt{x},\mathtt{y}) \to \mathtt{Id}_\mathtt{C}(\mathtt{x},\mathtt{y}) \;\equiv\; \mathtt{InDis}(\mathtt{x},\mathtt{y}) \to \mathtt{Id}_\mathtt{C}(\mathtt{x},\mathtt{y})$$

$$PII_-^+ :\equiv \mathtt{Alike}^-(\mathtt{x},\mathtt{y}) \to \mathtt{Id}_\mathtt{C}(\mathtt{x},\mathtt{y}) \;\equiv\; \neg\mathtt{Dis}(\mathtt{x},\mathtt{y}) \to \mathtt{Id}_\mathtt{C}(\mathtt{x},\mathtt{y})$$

$$PII_+^- :\equiv \mathtt{Alike}^+(\mathtt{x},\mathtt{y}) \to \neg\neg\mathtt{Id}_\mathtt{C}(\mathtt{x},\mathtt{y}) \;\equiv\; \mathtt{InDis}(\mathtt{x},\mathtt{y}) \to \neg\neg\mathtt{Id}_\mathtt{C}(\mathtt{x},\mathtt{y})$$

$$PII_-^- :\equiv \mathtt{Alike}^-(\mathtt{x},\mathtt{y}) \to \neg\neg\mathtt{Id}_\mathtt{C}(\mathtt{x},\mathtt{y}) \;\equiv\; \neg\mathtt{Dis}(\mathtt{x},\mathtt{y}) \to \neg\neg\mathtt{Id}_\mathtt{C}(\mathtt{x},\mathtt{y})$$

8.2.1   Entailments

The following entailments follow immediately from the relative strengths of the variants of material conditional involved:

$$
\begin{array}{ccc}
PII_-^+ & \longrightarrow & PII_+^+ \\
\downarrow & & \downarrow \\
PII_-^- & \longrightarrow & PII_+^-
\end{array}
$$

Since the conclusion of PII_+^- and PII_-^- is the rather weak $\neg\neg\mathtt{Id}_\mathtt{C}(\mathtt{x},\mathtt{y})$, these ought to be acceptable even to the opponent of PII. But perhaps they are too weak to be of interest. That leaves us with PII_+^+ and PII_-^+. If we replace classical implication and seek to avoid negation where possible, then we can take PII_+^+ to be the right version of PII in a constructive setting.

References

Steve Awodey.   Structuralism, Invariance, and Univalence. http://www.andrew.cmu.edu/user/awodey/preprints/siu.pdf, 2014.

John C. Baez and James Dolan. Categorification. In *Higher Category Theory*, number 230 in Contemp. Math., pages 1–36. American Mathematical Society, 1998.

Andrej Bauer.   The elements of an inductive type. http://math.andrej.com/2013/08/28/the-elements-of-an-inductive-type, August 2013.

Nicola Gambino, Chris Kapulkin, and Peter LeFanu Lumsdaine. The univalence axiom and functional extensionality. URL `http://www.pitt.edu/~krk56/UA_implies_FE.pdf`.

J. Ladyman and S. Presnell. Identity in Homotopy Type Theory, Part I: The Justification of Path Induction. *Philosophia Mathematica*, 2015. doi: 10.1093/philmat/nkv014. Available online at `http://philmat.oxfordjournals.org/content/early/2015/06/01/philmat.nkv014.short`.

James Ladyman and Stuart Presnell. Universes and univalence in homotopy type theory.

James Ladyman and Stuart Presnell. Does Homotopy Type Theory Provide a Foundation for Mathematics? *British Journal for the Philosophy of Science*, forthcoming. Available at `http://www.bristol.ac.uk/homotopy-type-theory/`.

Hannes Leitgeb and James Ladyman. Criteria of Identity and Structuralist Ontology. *Philosophia Mathematica*, 16(3):388–396, 2008.

nLab. Homotopy type system. `http://ncatlab.org/homotopytypetheory/show/Homotopy+Type+System`, 2014.

Stewart Shapiro. Identity, Indiscernibility, and ante rem Structuralism: The Tale of i and $-i$. *Philosophia Mathematica*, 16:285–309, 2008. doi: 10.1093/philmat/nkm042.

Mike Shulman. Universe polymorphism and typical ambiguity. Article at The n-Category Café, December 2012. URL `https://golem.ph.utexas.edu/category/2012/12/universe_polymorphism_and_typi.html`.

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

Vladimir Voevodsky. A type system with two kinds of identity types. `https://uf-ias-2012.wikispaces.com/file/view/HTS_slides.pdf/410105196/HTS_slides.pdf`, February 2013.