How do you identify one thing with another?

David Jaz Myers

Johns Hopkins University

May 14, 2019

Outline

1. What does it mean to identify one thing with another?
2. A formal definition of “identification”.

Outline

1. What does it mean to identify one thing with another?
2. But first a quick introduction to type theory.
3. A formal definition of “identification”.
4. Stating the *Univalence axiom*, which makes the type theoretic definition of “identification” work.

How do you identify one thing with another?

It depends what kind of things they are.

- To identify a **vector space** V with $\mathbb{R}^n$, it suffices to choose a basis $\{e_i\}$. We identify v in V with

$$(v^1, \ldots, v^n) \quad \text{where} \quad v \text{ is } v^1 e_1 + \cdots + v^n e_n.$$

- To identify the **fundamental group** $\pi_1(\mathbb{S}^1)$ of the circle with $\mathbb{Z}$, it suffices to choose a **generating loop** $\gamma : \mathbb{S}^1 \to \mathbb{S}^1$.

- To identify a **number** n with 3, we prove that n **equals** 3.

How things are identified matters

- Suppose that p is a point on a manifold M.
- Any chart U around p gives an identification of the tangent space T_pM with $\mathbb{R}^n$ (using coordinates).
- But any other chart V around p also gives an identification of T_pM with $\mathbb{R}^n$!
- Putting them together, we get a **transition matrix**

$$\mathbb{R}^n \xrightarrow{\text{from } U} T_pM \xrightarrow{\text{from } V} \mathbb{R}^n.$$

The ambiguity in how we identify T_pM with $\mathbb{R}^n$
is measured by the **group** $\mathbf{GL}_n(\mathbb{R})$.

What is Homotopy Theory?

Homotopy theory is the study of how things can be identified.the study of the algebraic structure of identification.

- In Algebraic Topology, an identification of one thing with another is a **continuous deformation** of the first into the second.

What is a Type Theory?

The more complicated the math gets, the more important it is **to keep track of where things live**.

- For a smooth function $f : \mathbb{R}^k \to \mathbb{R}^n$, we can make the Jacobian Jf matrix of its first partials and the Hessian Hf matrix of its second partials. But Jf represents a linear function while Hf represents a quadratic form.
- The unit circle $\mathbb{S}^1 \subseteq \mathbb{R}^2$ is contractible, but the unit circle $\mathbb{S}^1 \subseteq \mathbb{R}^1 - \{(0,0)\}$ is not.
- As an integer, 3 is not a unit. But as a rational number, it is.

Definition

A **type theory** is a formal system for keeping track of "where everything lives".

What is a Type Theory?

The more complicated the math gets, the more important it is keep track of what kind of thing everything is.

- For a smooth function $f : \mathbb{R}^k \to \mathbb{R}^n$, we can make the Jacobian Jf matrix of its first partials and the Hessian Hf matrix of its second partials. But Jf represents a linear function while Hf represents a quadratic form.
- The unit circle $\mathbb{S}^1 \subseteq \mathbb{R}^2$ is contractible, but the unit circle $\mathbb{S}^1 \subseteq \mathbb{R}^1 - \{(0,0)\}$ is not.
- As an integer, 3 is not a unit. But as a rational number, it is.

Definition

A **type theory** is a formal system for keeping track of what kind of thing everything is.

What is a Type Theory?

Definition

A **type theory** is a formal system for keeping track of what kind of thing everything is.

- $$a : A$$
  means that A is the kind of thing that the thing a is.
- Shorter: a *is of type* A.
- E.g.

$$3 : \mathbb{N}$$

$$\pi : \mathbb{R}$$

$$\mathbb{N} : \textbf{Set}$$

$$\mathbb{Z} : \textbf{Group}$$

Judgements

$$a : A$$

is *not* a "proposition" – it is not up for debate.

- Saying $3 : \mathbb{N}$ is a **judgement**: the fact that 3 is a *number* is just part of what we mean by 3.
- $3 : \mathbb{Z}$ and $3 : \mathbb{Q}$ are *different* 3*s*. For example, the second is a unit while the first is not.
- Similarly, we use "$a \equiv b$" to say that a is *judged* to be equal to b by *definition*. For example,

$$3 \equiv \mathbf{suc}(\mathbf{suc}(\mathbf{suc}(0))).$$

Dependent Types

A type can depend on a variable of another type.

- For example, given $k : \mathbb{N}$ the type $\{n : \mathbb{N} \mid n \geq k\}$ is a type which depends on k.
- The tangent space T_pM of a manifold M at a point $p : M$ is a type which depends on p.

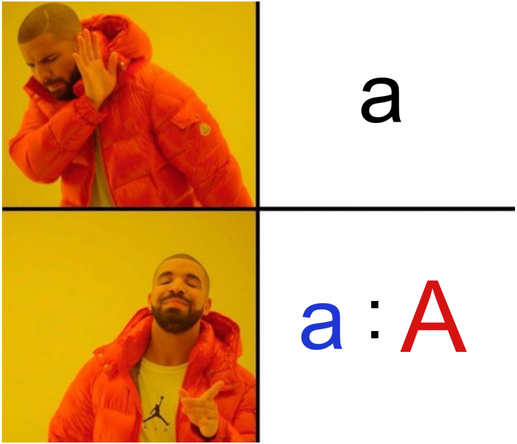
The codomain of a function can depend on its domain.

- The function $k \mapsto k + 1$ has type $(k : \mathbb{N}) \to \{n : \mathbb{N} \mid n \geq k\}$.
- A vector field is naturally a dependent function. A vector field assigns to each point $p : M$ of a manifold a vector $v_p : T_pM$ of its tangent space. This has type $v : (p : M) \to T_pM$.

Functions

Every thing is a certain kind of thing.

- In a type theory, every free variable must be annotated with its type.



- Given types A and B depending on A,

$$(a : A) \to B(a) \qquad \text{or, sometimes, } \Pi_{a:A} B(a)$$

Inductive Types: Natural Numbers

If a type A is an *inductive type*, we may assume that a free variable $a : A$ is of one several prescribed forms.

- We may assume a free natural number $n : \mathbb{N}$ is either of the form
  1. $n \equiv 0$, or
  2. $n \equiv \textbf{suc}(m)$ with $m : \mathbb{N}$.
- To define $+ : \mathbb{N} \to (\mathbb{N} \to \mathbb{N})$, we assume a free variable $n : \mathbb{N}$ and seek a function of type $\mathbb{N} \to \mathbb{N}$.
  1. If $n \equiv 0$, then we have $\text{id} \equiv x \mapsto x : \mathbb{N} \to \mathbb{N}$, or
  2. If $n \equiv \textbf{suc}(m)$, then we have $x \mapsto \textbf{suc}(x + m) : \mathbb{N} \to \mathbb{N}$

  In total, we have

$$n \mapsto \begin{cases} x \mapsto x & \text{if } n \equiv 0 \\ x \mapsto \textbf{suc}(x + m) & \text{if } n \equiv \textbf{suc}(m). \end{cases} : \mathbb{N} \to (\mathbb{N} \to \mathbb{N})$$

The Type of Identifications

Given any two terms $a, b : A$, we have a **type** $a =_A b$ **of identifications of** a **with** b.

- We may assume that free variables $b : A$ and $p : a =_A b$ are of the form
  1. $\mathsf{refl} : a =_A a$.
- To define **sym** : $(a, b : A) \to a =_A b \to b =_A a$, assume that $a, b : A$ and $p : a =_A b$ are free variables.
  1. If $b \equiv a$ and $p \equiv \mathsf{refl}$, then $\mathsf{refl} : b =_A a$.

  So,

$$a, b, p \mapsto \begin{cases} \mathsf{refl} & \text{if } p \equiv \mathsf{refl} \end{cases} : (a, b : A) \to a =_A b \to b =_A a.$$

Hmmm...

Question

Given that elements $p : a =_A b$ have only one prescribed form, is there at most one element of type $a =_A b$ (namely, refl when $a \equiv b$)?

Pairs and Equivalences

Given a type A and a type B depending on A, we can form the type

$$(a : A) \times B(a) \qquad \text{or sometimes } \Sigma_{a:A} B(a)$$

whose elements are pairs $(a, b) : (a : A) \times B(a)$ where $a : A$ and $b : B(a)$.

Definition

A function $e : A \to B$ is an *equivalence* if there are functions $\ell, r : B \to A$ and identifications $p : \mathrm{id}_A =_{A \to A} \ell \circ e$ and $q : e \circ r =_{B \to B} \mathrm{id}_B$. In other words

$$e \text{ is an equivalence} :\equiv$$

$$(\ell : B \to A) \times (r : B \to A) \times (\mathrm{id}_A =_{A \to A} \ell \circ e) \times (e \circ r =_{B \to B} \mathrm{id}_B)$$

and

$$A \simeq B :\equiv (e : A \to B) \times e \text{ is an equivalence}$$

Univalence

Every identification p of a type A with a type B gives an equivalence id-to-equiv(p) : $A \simeq B$.

- How do we define the function id-to-equiv : $(A, B : \mathbf{Type}) \to A =_{\mathbf{Type}} B \to A \simeq B$?
- Assume that A and B are free variables of type Type, and that p : $A =_{\mathbf{Type}} B$.
  1. Since B and p are free, we may assume $B \equiv A$ and $p \equiv$ refl. Then id : $A \simeq B$ is an equivalence.

So,

$$\text{id-to-equiv} :\equiv A, B, p, \mapsto \begin{cases} \text{id} & \text{if } p \equiv \text{refl} \end{cases}$$

Univalence

The **Univalence Axiom** says that id-to-equiv : $A =_{\textbf{Type}} B \to A \simeq B$ is an equivalence. In other words,

$$\text{ua} : \text{id-to-equiv is an equivalence}$$

We may identify the type A with the type B by giving an equivalence $e : A \simeq B$.

Univalence implies that the formal definition of "identification" gives what we expect:

- If V : **VectorSpace**, then $V =_{\textbf{VectorSpace}} \mathbb{R}^n$ is the type of bases of V with n elements.
- If G : **Group**, then $G =_{\textbf{Group}} \mathbb{Z}$ is the type of isomorphisms of G with $\mathbb{Z}$.
- If $n : \mathbb{N}$, then $n =_{\mathbb{N}} 3$ has at most one element. To write down an element $e : n =_{\mathbb{N}} 3$ is the same as *proving* that n equals 3.