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”.

But first a quick introduction to type theory.

The **Univalence axiom**, which makes the type theoretic definition of “identification” work.
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

  \[(v^1, \ldots, v^n) \text{ where } v = v^1e_1 + \cdots + v^ne_n.\]

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

- To identify a **number** \( n \) with \( 3 \), we prove that \( n \text{ equals } 3 \).
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 $GL_n(\mathbb{R})$. 

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 $J_f$ matrix of its first partials and the Hessian $H_f$ matrix of its second partials. But $J_f$ represents a linear function while $H_f$ represents a quadratic form.
- The unit circle $S^1 \subseteq \mathbb{R}^2$ is contractible, but the unit circle $S^1 \subseteq \mathbb{R}^1 - \{(0, 0)\}$ 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 $S^1 \subseteq \mathbb{R}^2$ is contractible, but the unit circle $S^1 \subseteq \mathbb{R}^1 - \{(0, 0)\}$ 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.

  \[
  3 : \mathbb{N} \\
  \pi : \mathbb{R} \\
  \mathbb{N} : \text{Set} \\
  \mathbb{Z} : \text{Group}
  \]
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 3s. 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,

\[
3 \equiv \text{suc}(\text{suc}(\text{suc}(0))).
\]
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 $A$,

$$ (a : A) \rightarrow B(a) \quad \text{or, sometimes, } \prod_a A B(a) $$
Inductive Types: Natural Numbers

If a type \( A \) is an \textit{inductive type}, we may assume that a free variable \( a : A \) is of one several prescribed forms.

1. We may assume a free natural number \( n : \mathbb{N} \) is either of the form
   1. \( n \equiv 0 \), or
   2. \( n \equiv \text{suc}(m) \) with \( m : \mathbb{N} \).

2. 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 \( n \equiv \text{suc}(m) \), then we have \( x \mapsto \text{suc}(x + m) : \mathbb{N} \to \mathbb{N} \)

In total, we have

\[
\begin{align*}
\text{n} & \mapsto \begin{cases} 
\text{x} & \mapsto \text{x} & \text{if } n \equiv 0 \\
\text{x} & \mapsto \text{suc}(x + m) & \text{if } n \equiv \text{suc}(m).
\end{cases} : \mathbb{N} \to (\mathbb{N} \to \mathbb{N})
\end{align*}
\]
The Type of Identifications

Given any two terms \( a, b : A \), we have a **type** \( a \equiv_A b \) **of identifications** of \( a \) with \( b \).

- We may assume that free variables \( b : A \) and \( p : a =_A b \) are of the form
  \[
  \text{refl} : a =_A a.
  \]

- To define \( \text{sym} : (a, b : A) \rightarrow a =_A b \rightarrow b =_A a \), assume that \( a, b : A \) and \( p : a =_A b \) are free variables.
  
  If \( b \equiv a \) and \( p \equiv \text{refl} \), then \( \text{refl} : b =_A a \).

So,

\[
a, b, p \mapsto \begin{cases} 
\text{refl} & \text{if } p \equiv \text{refl} : (a, b : A) \rightarrow a =_A b \rightarrow b =_A a.
\end{cases}
\]
Hmmm...

**Question**

Given that elements \( p : a \equiv_A b \) have only one prescribed form, is there at most one element of type \( a \equiv_A b \) (namely, \texttt{refl} when \( a \equiv b \))?
Given a type $A$ and a type $B$ depending on $A$, we can form the type

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

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 $p : \text{id}_A \equiv_{A \to A} \ell \circ e$ and $q : e \circ r \equiv_{B \to B} \text{id}_B$. In other words

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

$$(\ell : B \to A) \times (r : B \to A) \times (\text{id}_A \equiv_{A \to A} \ell \circ e) \times (e \circ r \equiv_{B \to B} \text{id}_B)$$

and

$$A \simeq B :\equiv (e : A \to B) \times e \text{ is an equivalence}$$
Every identification $p$ of a type $A$ with a type $B$ gives an equivalence $\text{id-to-equiv}(p) : A \simeq B$.

- How do we define the function $\text{id-to-equiv} : (A, B : \text{Type}) \rightarrow A =_{\text{Type}} B \rightarrow A \simeq B$?
- Assume that $A$ and $B$ are free variables of type $\text{Type}$, and that $p : A =_{\text{Type}} B$.

  Since $B$ and $p$ are free, we may assume $B \equiv A$ and $p \equiv \text{refl}$. Then $\text{id} : A \simeq B$ is an equivalence.

So,

$$\text{id-to-equiv} : \equiv A, B, p, \mapsto \begin{cases} \text{id} & \text{if } p \equiv \text{refl} \end{cases}$$
The **Univalence Axiom** says that \( \text{id-to-equiv} : A \equiv Type B \rightarrow A \simeq B \) is an equivalence. In other words,

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

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 \( V : \text{VectorSpace} \), then \( V =_{\text{VectorSpace}} \mathbb{R}^n \) is the type of bases of \( V \) with \( n \) elements.
- If \( G : \text{Group} \), then \( G =_{\text{Group}} \mathbb{Z} \) 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.