THE TWO-SIDED IDENTITY TYPE OF MARTIN-LÖF TYPE THEORY INTRODUCTION TO TYPE THEORIES, OPLSS 2025

ANJA PETKOVIĆ KOMEL

The *identity type*, also called *propositional equality type* is one of the most important types in Martin-Löf type theory. It allows us to reason about equality from the *propositions-as-types* perspective. The idea of the identity type is that given a type A and elements $a : A$ and $b : A$, the equality $a =_A b$ should also be a (dependent) type on a and b. Its elements are witnesses that a and b are equal, so $p : a =_A b$ is *an identification* of a and b. We form it as an inductive type. There are several ways to construct the identity type. Unlike [1] that introduced the identity type one-sidedly to showcase its connection to homotopy theory, we will use a more standard presentation of the identity type, the two-sided identity type.

1. The inductive definition of the two-sided identity type

The two-sided identity type is given by the following rules:

(1) **The formation rule**:
$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : A}{\Gamma \vdash a =_A b \ \mathsf{type}}$$
(2) **The introduction rule**:
$$\frac{\Gamma \vdash a : A}{\Gamma \vdash \mathsf{refl}_a : a =_A a}$$
(3) **The elimination rule (J-rule)**, the induction principle:
$$\frac{\Gamma, x : A, y : A, p : x =_A y \vdash P(x, y, p) \ \mathsf{type}}{\Gamma \vdash J : (\Pi_{x:A} P(x, x, \mathsf{refl}_x)) \to \Pi_{a:A} \Pi_{b:A} \Pi_{q:a=_A b} P(a, b, q)}$$
(4) **The computation rule**:
$$\frac{\Gamma, x : A, y : A, p : x =_A y \vdash P(x, y, p) \ \mathsf{type} \qquad \Gamma \vdash d : \Pi_{x:A} P(x, x, \mathsf{refl}_x) \qquad \Gamma \vdash a : A}{\Gamma \vdash J(d, a, a, \mathsf{refl}_a) \doteq d(a) : P(a, a, \mathsf{refl}_a)}$$

The idea of the induction principle for the identity type is that in order to define a function *out of* the identity type, we only need to give it on refl (the "diagonal" d in the computation rule).

Remark 1. We will omit the index A at the type $a =_A b$ for clarity, when it can be inferred from a and b.

2. Properties of the identity type

By definition, the identity type is *reflexive*, i.e. we have an element $\mathsf{refl}_a : a =_A a$ for every A type and $a : A$. Following the section in the groupidal structure of types (Section 5.2) in [1] we note the following properties of the identity types.

We start by what we can understand as the *transitivity of the propositional equality.*

Date: June 2025.

Definition 2. Let A be a type. We define the **concatenation** operation

$$\mathsf{concat} : \Pi_{x,y,z:A}(x = y) \to ((y = z) \to (x = z)).$$

We will write $p \centerdot q$ for $\mathsf{concat}(p, q)$.

We will of course use the J-rule to construct the concat operation. We first observe that we can swap the arguments and instead construct a function

$$g : \Pi_{x,y:A}(x = y) \to \Pi_{z:A}((y = z) \to (x = z))$$

And then we can define

$$\mathsf{concat}(x, y, z, p, q) := g(x, y, p, z, q).$$

We now use the J-rule for $P(x, y, q) := \Pi_{z:A}((y = z) \to (x = z))$. We need to provide an element of $P(x, x, \mathsf{relf}_x) \doteq \Pi_{z:A}((x = z) \to (x = z))$, for which we can use the identity function $\lambda z.\ \mathsf{id}_{(x=z)}$. We conclude the construction with

$$g := J(\lambda x.\ \lambda z.\ \mathsf{id}_{(x=z)}).$$

Propositional equality is also *symmetric* in the following sense.

Definition 3. Let A be a type. We define the **inverse operation**

$$\mathsf{inv} : \Pi_{x,y:A}(x = y) \to (y = x).$$

We write p^{-1} for $\mathsf{inv}(p)$.

We now adopt a more informal style using the induction principle of the identity type without explicitly applying the J-rule in the constructions. Since the J-rule allows us to define a (dependent) function out of the identity type by just providing the term for the refl case, we can construct the function inv by giving $\mathsf{inv}(\mathsf{refl}_x) := \mathsf{refl}_x$.

We allow ourselves this informal style because it is more readable and also because it is the way we use the induction principle (J-rule) in proof assistants (Agda, Rocq). The constructions for other properties mentioned in [1] proceed the same: concatenation is associative, refl acts as a unit for concatenation operation, the inverse laws hold etc. We can also define action on paths ap (for $f : A \to B$) and transport tr (for B a type family over A) in the same way:

$$\mathsf{ap}_f : \Pi_{x,y:A}(x =_A y) \to (f(x) =_B f(y)) \qquad \mathsf{tr}_B : \Pi_{x,y:A}(x =_A y) \to (B(x) \to B(y))$$
$$\mathsf{ap}_f(x, x, \mathsf{refl}_x) = \mathsf{refl}_{f(x)} \qquad \mathsf{tr}_B(x, x, \mathsf{refl}_x) = \mathsf{id}_{B(x)}$$

3. Judgemental vs. propositional equality

Now that we have two notions of equality, the question is how do these two interact? We think of judgemental equality as the one that "computes" and falls under the reflexivity case for the propositional equality. In fact proof assistants Agda and Coq implement judgemental equality precisely in this way. We can capture this behavior in the following derivable rule:

$$\frac{\Gamma \vdash p : a =_A b \qquad \Gamma \vdash a \doteq a' : A}{\Gamma \vdash p : a' =_A b}$$

with derivation

$$\frac{\Gamma \vdash p : a =_A b \qquad \dfrac{\Gamma \vdash a \doteq a' : A \qquad \Gamma \vdash b \doteq b : A}{\Gamma \vdash (a =_A b) \doteq (a' =_A b)}}{\Gamma \vdash p : a' =_A b}$$

where we used the congruence rule for propositional equality in the right branch.

References

[1] Egbert Rijke. Introduction to homotopy type theory. Available on arXiv: 2212.11082, 2022.