Outline:

Last time: inductive types – identity types } §3–4 Rijke

This time: identity type } §5 Rijke

→ homotopy

Why do we need the identity type? (if we're not interested in homotopy)

We already have a notion of equality:

$\doteq$ judgmental equality

(The identity type is called propositional equality.)

Logical interpretation: types are propositions / terms are proofs

→ Proving equality means constructing a term of an equality type

We can prove many judgmental equalities:

Ex. $\mathrm{add}(x,0) \doteq x$

$\mathrm{add}(x, sy) \doteq s\ \mathrm{add}(x,y)$

... but not all the ones we want.

Ex. $\mathrm{add}(0, x) \doteq x$

$\mathrm{add}(sx, y) \doteq s\ \mathrm{add}(x,y)$

$\mathbb{N}$-elim:
(aka induction)

$$\frac{\begin{array}{c} x:\mathbb{N} \vdash D(x) \text{ type} \\ \vdots \end{array}}{x:\mathbb{N} \vdash \mathrm{ind} : D(x)}$$

Type constructors often internalize structure

- At a 'meta' level, we can talk about contexts:

  Ex. $x:A,\ y:B(x),\ z:C(x,y) \vdash$

We can discuss this at the 'type-and-term level' using Σ-types:

Ex. $z : \sum_{x:A} \sum_{y:B(x)} C(x,y)$

- Similarly, we can talk about dependent terms as 'meta' function:

Ex. $x : A,\ y : B(x) \vdash c(x,y) : C(x,y)$

We can internalize such 'meta'-functions as terms of a Π-type

Ex. $c : \prod_{x:A} \prod_{y:B(x)} C(x,y)$

$\left.\begin{array}{l} \bullet\ \text{bool} \\ \bullet\ \mathbb{N} \\ \bullet\ \emptyset \\ \bullet\ \mathbb{1} \end{array}\right\}$ can be seen as internalizing external notions

- The universe type internalizes the judgment A type.

- We'll see that the identity type internalizes judgmental equality.

Identity type =

NB: Compare these with the rules in Rijke.
"based path induction"

$=$-form $\quad \dfrac{\ulcorner\vdash A \text{ type} \quad \ulcorner\vdash a : A \quad \ulcorner\vdash b : A}{a =_A b \text{ type}}$

$=$-intro $\quad \dfrac{\ulcorner\vdash a : A}{\ulcorner\vdash r_a : a =_A a}$

(reflexivity)

$=$-elim $\quad \dfrac{x : A,\ y : A,\ z : x =_A y \vdash D(x,y,z) \text{ type} \qquad x : A \vdash d : D(x,x,r_x)}{x : A,\ y : A,\ z : x =_A y \vdash \mathrm{ind}_=(d,x,y,z) : D(x,y,z)}$

$=$-comp $\quad \dfrac{x : A,\ y : A,\ z : x =_A y \vdash D(x,y,z) \text{ type} \qquad x : A \vdash d : D(x,x,r_x)}{x : A \vdash \mathrm{ind}_=(d,x,x,r_x) \doteq d : D(x,x,r_x)}$

Type constructors internalize structure

- We can talk about judgmental equality at a 'meta' level.

  Ex. $a \doteq b : A$

We can internalize this using identity types.

Ex. $r_a : a =_A b$

$$\begin{cases} \text{If } a \doteq b : A, \text{ then } (a =_A b) \doteq (a =_A a), \text{ and} \\ \text{if } r_a : a =_A a, \text{ then } r_a : a =_A b. \end{cases}$$

→ Reflexity (r_-) turns judgmental equalities into propositional equalities.

Functoriality

Functions act on paths / terms of the identity type.

Prop. For any two types A, B, any function $f : A \to B$, and any two terms $a, a' : A$, there is a function

$$\mathrm{ap}_f : a =_A a' \longrightarrow fa =_B fa'$$

NB. Every proposition we make in type theory is really a type, but we often write them in English, at least partially.

This propositson stands for

$$\prod_{A,B:\mathrm{Type}} \ \prod_{f:A\to B} \ \prod_{a,a':A} a =_A a' \longrightarrow fa =_B fa'$$

Functoriality: $\quad \mathrm{ap} : \prod_{f:A\to B} \ \prod_{a,a':A} a =_A a' \longrightarrow fa =_B fa'$

$$\frac{f : A \to B,\ a : A \vdash r_{fa} : fa =_B fa}{f : A \to B,\ a, a' : A,\ p : a =_A a' \vdash \mathrm{ind}_=(r_f, a, a', p) : fa =_B fa'}$$

$$\lambda f. \lambda a, a'. \lambda p. \mathrm{ind}_=(r_f, a, a', p) : \prod_{f : A \to B} \prod_{a, a' : A} a =_A a' \to fa =_B fa'$$

<u>Example</u>. $\prod_{n : \mathbb{N}} \mathrm{add}(0, n) = n$

Use: $\mathrm{add}(n, 0) \doteq n$

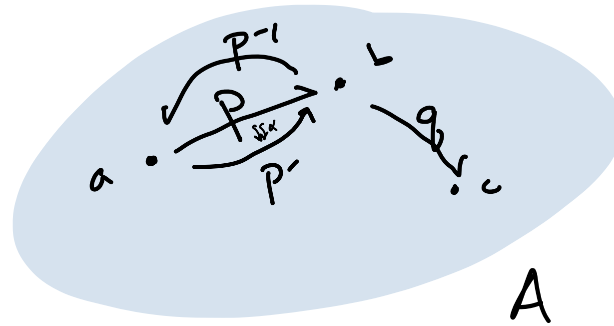
$\mathrm{add}(n, sm) \doteq s\, \mathrm{add}(n, m)$

$=$-elim
$$\frac{\Gamma,\ x : A,\ y : A,\ z : x =_A y \vdash D(x, y, z) \text{ type} \qquad \Gamma,\ x : A \vdash d : D(x, x, r_x)}{\Gamma,\ x : A,\ y : A,\ z : x =_A y \vdash \mathrm{ind}_=(d, x, y, z) : D(x, y, z)}$$

$$\frac{\dfrac{r_0 : \mathrm{add}(0, 0) = 0 \qquad n : \mathbb{N},\ p : \mathrm{add}(0, n) = n \vdash \mathrm{ap}_s p : \mathrm{add}(0, sn) = sn}{n : \mathbb{N} \vdash \mathrm{ind}_{\mathbb{N}}(r_0, \mathrm{ap}_s, n) : \mathrm{add}(0, n) = n}}{\lambda n. \mathrm{ind}_{\mathbb{N}}(r_0, \mathrm{ap}_s, n) : \prod_{n : \mathbb{N}} \mathrm{add}(0, n) = n}$$

The groupoidal behavior of types

(the first homotopical phenomena)



We can think of types as consisting of points (terms) connected by homotopies/paths (identities).

We can:

- have multiple equalities of the same type (e.g. $p, p' : a =_A b$)
- take the inverse of an identity/path (e.g. $p^{-1} : b =_A a$)
- take the composition of two paths (e.g. $p \cdot q : a =_A c$)
- have equalities of equalities (e.g. $\alpha : p =_{a =_A b} p'$)

This is how homotopies in spaces behave.

The space interpretation.

Thm (Voevodsky) There is an interpretation of dependent type theory into Spaces (the category of Kan complexes) in which

types $\rightsquigarrow$ spaces or Kan complexes

terms $\rightsquigarrow$ points or 0-cells

equalities $\rightsquigarrow$ paths or 0-cells of the path object

$\pi : \sum_{b:B} E(b) \to B \rightsquigarrow$ or Kan fibrations

$\prod_{b:B} E(b) \rightsquigarrow$ or the space of sections of $\pi : \sum_{b:B} E(b) \to B$

Inverse of equalities : $\prod_{a,b:A} a =_A b \to b =_A a$

$$\cfrac{\cfrac{a:A \vdash r_a : a =_A a}{a,b:A,\ p: a =_A b \vdash \mathrm{ind}_=(r,a,b,p) : b =_A a}}{\lambda a,b,p.\ \mathrm{ind}_=(r,a,b,p) : \prod_{a,b:A} a =_A b \to b =_A a}$$

Composition of equalities : $\prod_{a,b,c:A} a =_A b \longrightarrow b =_A c \longrightarrow a =_A c$

=-elim

$$\frac{x:A,\ y:A,\ z:x=_A y \vdash D(x,y,z)\ \text{type} \qquad x:A \vdash d:D(x,x,r_x)}{x:A,\ y:A,\ z:x=_A y \vdash \mathrm{ind}_=(d,x,y,z):D(x,y,z)}$$

$$\cfrac{\cfrac{\cfrac{c:A,\ a:A \vdash \lambda x.x : a=_A c \to a=_A c}{c:A,\ a:A,\ b:A,\ p:a=_A b \vdash \mathrm{ind}_=(\lambda x.x, a, b, p) : b=_A c \to a=_A c}}{a,b,c:A,\ p:a=_A b \vdash \mathrm{ind}_=(\lambda x.x, a, b, p) : b=_A c \to a=_A c}}{\lambda a,b,c.\ \mathrm{ind}_=(\lambda x.x, a, b, p) : \prod_{a,b,c:A} a=_A b \to b=_A c \to a=_A c}$$

Transport:

Prop. For any dependent type $x:B \vdash E(x)$ type, any terms $b, b':B$ and any equality $p: b =_B b'$, there is a function

$$tr_{B,p}: E(b) \to E(b')$$

- This ensures that everything respects equality.

- This is part of a more sophisticated relationship between homotopy theory (Quillen model category theory) and type theory. Transport says that $\pi : \sum_{b:B} E(b) \to B$ behave like fibrations in a QMC.

<u>Transport</u>. $\prod_{b,b':B} (b =_B b') \to E(b) \longrightarrow E(b')$

$$\cfrac{\cfrac{b : B \vdash \lambda x.x \quad : \quad E(b) \to E(b)}{b : B,\, b' : B,\, p : b =_B b' \vdash \mathrm{ind}_=(\lambda x.x,\, b, b', p) : E(b) \to E(b')}}{\lambda b, b', p.\, \mathrm{ind}_=(\lambda x.x, b, b', p) : \prod_{b,b':B} (b =_B b') \to E(b) \longrightarrow E(b')}$$

<u>The homotopical content so far</u>

- Types behave like spaces.
- However, the UIP (the principle of uniqueness of identity proofs) is consistent with what we've done so far.

$$\mathrm{UIP} := \prod_{a,b:A} \prod_{p,q : a =_A b} p =_{a =_A b} q$$

- i.e., we still have an interpretation into Set.

- Only with higher inductives types and the univalence axiom, honest homotopical behaviour occurs.

- i.e., we won't have an interpretation into sets.