

Outline :

Last time: inductive types - identity type } §3-4
Rijke

This time: identity type } §5
→ homotopy

Why do we need the identity type?

(If we're not interested in homotopy.)

We already have a notion of equality:

judgmental equality \doteq

(The identity type is called propositional equality $=$.)

Logical interpretation: propositions are types / proofs are terms.

To prove an equality (and be consistent with the logical interpretation)
we want to produce a term of a type of equalities.

Why do we need the identity type actually?

We can prove many judgmental equalities using computation rules...

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

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

Why do we need the identity type actually?

We can prove many judgmental equalities using computation rules...

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

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

... but not all the equalities we want!

Ex. One cannot prove $\text{add}(n, 0) \doteq n$.

To prove this, we need to induct on n (i.e. use \mathbb{N} -elimination),
but this only allows us to construct a term of a type.

We will be able to prove $\text{add}(n, 0) = n$.

\mathbb{N} -elim (roughly):

$$\frac{n : \mathbb{N} \vdash D(n) \text{ type}}{n : \mathbb{N} \vdash \text{ind}_{\mathbb{N}}(n) : D(n)}$$

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 by using Σ -types:

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

Type constructors often internalize structure

- At a 'meta' level, we can talk about dependent terms as functions:

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

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

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

Type constructors often internalize structure

- bool
 - \mathbb{N}
 - \emptyset
 - $\mathbb{1}$
- } can also be seen as internalizing
external versions.
- The universe type (next lecture) internalizes the judgment of the form
 $A \text{ type}$
 - We'll see how the identity type internalizes judgmental equality ...

Identity type =

= - form

$$\frac{A \text{ type } a : A \ b : A}{a =_A b \text{ type}}$$

= - intro

$$\frac{a : A}{r_a : a =_A a}$$

= - elim

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

= - comp

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

Identity type =

= - form

$$\frac{\Gamma \vdash A \text{ type } r:a:A \ r:b:A}{\Gamma \vdash a =_A b \text{ type}}$$

(I am omitting the extra context Γ everywhere for readability, but it's still there.)

= - intro

$$\frac{\Gamma \vdash a : A}{\Gamma \vdash r_a : a =_A a}$$

= - elim

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

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

= - comp

Identity type =

= - form

$$\frac{A \text{ type } a : A \ b : A}{a =_A b \text{ type}}$$

= - intro

$$\frac{a : A}{r_a : a =_A a}$$

= - elim

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

= - comp

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

Identity type = with based path induction

\equiv^b form

$$\frac{A \text{ type } a : A}{x : A \vdash a =_A x \text{ type}}$$

$$\equiv \text{- form} \quad \frac{A \text{ type } a : A \ b : A}{a =_A b \text{ type}}$$

\equiv^- intro

$$\frac{a : A}{r_a : a =_A a}$$

\equiv^b - elim

$$\frac{\begin{array}{c} a : A \\ x : A, z : a =_A x \vdash D(x, z) \text{ type} \\ \vdash d : D(a, r_a) \end{array}}{x : A, z : a =_A x \vdash \text{ind}_=(d, x, z) : D(x, z) \text{ type}}$$

$$\equiv^- \text{- elim} \quad \frac{\begin{array}{c} x : A, y : A, z : x =_A y \vdash D(x, y, z) \text{ type} \\ x : A \vdash d : D(x, x, r_x) \end{array}}{x : A, y : A, z : x =_A y \vdash \text{ind}_=(d, x, y, z) : D(x, y, z) \text{ type}}$$

$$x : A, z : a =_A x \vdash \text{ind}_=(d, x, z) : D(x, z) \text{ type}$$

\equiv^b - comp

$$\frac{x : A, z : x =_A y \vdash D(x, z) \text{ type} \\ \vdash d : D(a, r_a)}{\vdash \text{ind}_=(d, a, r_a) \doteq d : D(a, r_a)}$$

$$\equiv^- \text{- comp} \quad \frac{\begin{array}{c} x : A, y : A, z : x =_A y \vdash D(x, y, z) \text{ type} \\ x : A \vdash d : D(x, x, r_x) \end{array}}{x : A \vdash \text{ind}_=(d, x, x, r_x) \doteq d : D(x, x, r_x)}$$

$$\vdash \text{ind}_=(d, a, r_a) \doteq d : D(a, r_a)$$

Type constructors often internalize structure

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

Ex. $a \doteq b : A$

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

Ex. $r_a : a =_A b$

Note that the rules governing equality say that

if $a \doteq b : A$, then $(a =_A a) \doteq (a =_A b)$, and

if $r_a : a =_A a$ and $(a =_A a) \doteq (a =_A b)$, then $r_a : a =_A b$.

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

Functionality

Functions act on paths.

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

$$ap_f : a =_A a' \rightarrow fa =_B fa'$$

NB. Every proposition we make in type theory is really a type, but we often write some of the type in words for ease of understanding.

This proposition stands for

$$\prod_{A,B:\text{Type}} \prod_{f:A \rightarrow B} \prod_{a,a':A} a =_A a' \rightarrow fa =_B fa'.$$

Functionality:

$$ap : \prod_{f:A \rightarrow B} \prod_{a,a':A} a=_A a' \rightarrow fa =_B fa'$$

= - elim

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

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

Functionality:

$$ap : \prod_{f:A \rightarrow B} \prod_{a,a':A} \frac{a =_A a'}{f_a =_B f_{a'}}$$

$$? : \prod_{f:A \rightarrow B} \prod_{a,a':A} \frac{a =_A a'}{f_a =_B f_{a'}}$$

= - elim

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

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

Functionality: $\frac{ap : \prod_{f:A \rightarrow B} \prod_{a,a':A} a=_A a' \rightarrow fa =_B fa'}{}$

$$\frac{f: A \rightarrow B, a, a': A, p: a=_A a' \vdash ? \quad : fa =_B fa'}{}$$

$$? : \frac{\prod_{f:A \rightarrow B} \prod_{a,a':A} a=_A a' \rightarrow fa =_B fa'}{}$$

= - elim

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

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

Functionality: $\frac{ap : \prod_{f:A \rightarrow B} \prod_{a,a':A} a=_A a' \rightarrow fa =_B fa'}{}$

$$\frac{f : A \rightarrow B, a \vdash ? : fa =_B fa}{f : A \rightarrow B, a, a' : A, p : a =_A a' \vdash ? : fa =_B fa'}$$

? : $\prod_{f:A \rightarrow B} \prod_{a,a':A} a=_A a' \rightarrow fa =_B fa'$

= - elim

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

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

Functionality: $\frac{ap : \prod_{f:A \rightarrow B} \prod_{a,a':A} a=_A a' \rightarrow fa =_B fa'}{}$

$$\frac{f : A \rightarrow B, a \vdash r_{fa} : fa =_B fa}{f : A \rightarrow B, a, a' : A, p : a =_A a' \vdash ? : fa =_B fa'}$$

$$? : \frac{\prod_{f:A \rightarrow B} \prod_{a,a':A} a=_A a' \rightarrow fa =_B fa'}{}$$

= - elim

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

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

Functionality: $\text{ap} : \prod_{f:A \rightarrow B} \prod_{a,a':A} a =_A a' \rightarrow fa =_B fa'$.

$$f : A \rightarrow B, a \vdash r_{fa} : fa =_B fa$$

$$f : A \rightarrow B, a, a' : A, p : a =_A a' \vdash \text{ind}_=(a, r_{fa}, a, a', p) : fa =_B fa'$$

$$\text{?} : \prod_{f:A \rightarrow B} \prod_{a,a':A} a =_A a' \rightarrow fa =_B fa'$$

= - elim

$$x : A, y : A, z : x =_A y \vdash D(x, y, z) \text{ type}$$

$$x : A \vdash d : D(x, x, r_x)$$

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

Functionality: $\frac{ap : \prod_{f:A \rightarrow B} \prod_{a,a':A} a=_A a' \rightarrow fa =_B fa'}{}$

$$\frac{f : A \rightarrow B, a + r_{fa} : fa =_B fa}{}$$

$$\frac{f : A \rightarrow B, a, a' : A, p : a =_A a' \vdash \text{ind}_=(a.r_{fa}, a, a', p) : fa =_B fa'}{}$$

$$\lambda f. \lambda a. \lambda a'. \text{ind}_=(a.r_{fa}, a, a', p) : \prod_{f:A \rightarrow B} \prod_{a,a':A} a=_A a' \rightarrow fa =_B fa'$$

= - elim

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

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

Example: $\underset{n:N}{\text{TT}} \text{ add}(0, n) = n$

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

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

Example: $\prod_{n:N} \text{add}(0, n) = n$

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

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

?

: $\prod_{n:N} \text{add}(0, n) = n$

Example: $\prod_{n:\mathbb{N}} \text{add}(0, n) = n$

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

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

$$\frac{n : \mathbb{N} \vdash ? \quad : \quad \text{add}(0, n) = n}{? \quad : \quad \prod_{n:\mathbb{N}} \text{add}(0, n) = n}$$

Example: $\prod_{n:N} \text{add}(0, n) = n$

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

$\text{add}(n, sn) \doteq s \text{ add}(n, m)$

? : $\text{add}(0, 0) = 0$ $n:N, p: \text{add}(0, n) = n \vdash ? : \text{add}(0, sn) = sn$

$n:N \vdash ? : \text{add}(0, n) = n$

? : $\prod_{n:N} \text{add}(0, n) = n$

Example: $\prod_{n:N} \text{add}(0, n) = n$

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

$\text{add}(n, sn) \doteq s \text{ add}(n, m)$

$r_0: \text{add}(0, 0) = 0$

$n:N, p: \text{add}(0, n) = n \vdash \text{ap}_s p: \text{add}(0, sn) = sn$

$n:N \vdash ? : \text{add}(0, n) = n$

$? : \prod_{n:N} \text{add}(0, n) = n$

Example: $\prod_{n:N} \text{add}(0, n) = n$

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

$\text{add}(n, sn) \doteq s \text{ add}(n, m)$

$r_0: \text{add}(0, 0) = 0$

$n:N, p: \text{add}(0, n) = n \vdash ap_s p: \text{add}(0, sn) = sn$

$n:N \vdash \text{ind}_N(r_0, ap_s, n) : \text{add}(0, n) = n$

?

$: \prod_{n:N} \text{add}(0, n) = n$

Example: $\prod_{n:N} \text{add}(0, n) = n$

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

$\text{add}(n, sn) \doteq s \text{ add}(n, m)$

$r_0 : \text{add}(0, 0) = 0$

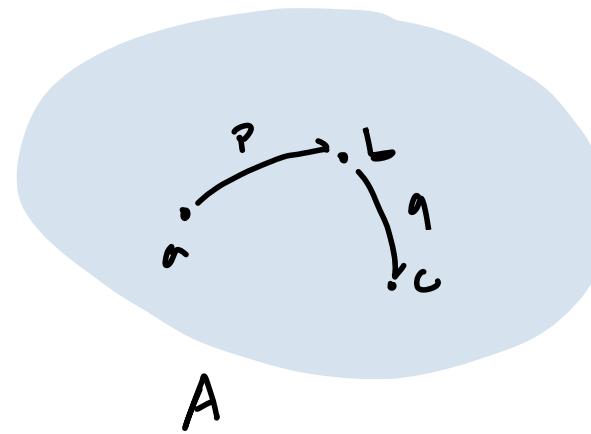
$n:N, p : \text{add}(0, n) = n \vdash ap_s p : \text{add}(0, sn) = sn$

$n:N \vdash \text{ind}_N(r_0, ap_s, n) : \text{add}(0, n) = n$

$\lambda n. \text{ind}_N(r_0, ap_s, n) : \prod_{n:N} \text{add}(0, n) = n$

The groupoidal behaviour of types

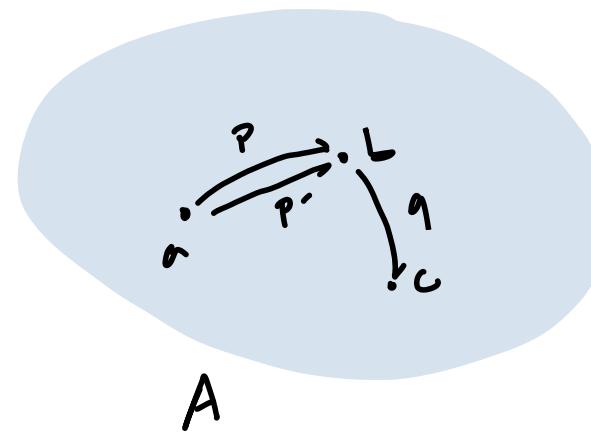
(The first homotopical phenomena)



We can now think of types as collections of points (terms) connected by homotopies/paths (equalities).

The groupoidal behaviour of types

(The first homotopical phenomena)



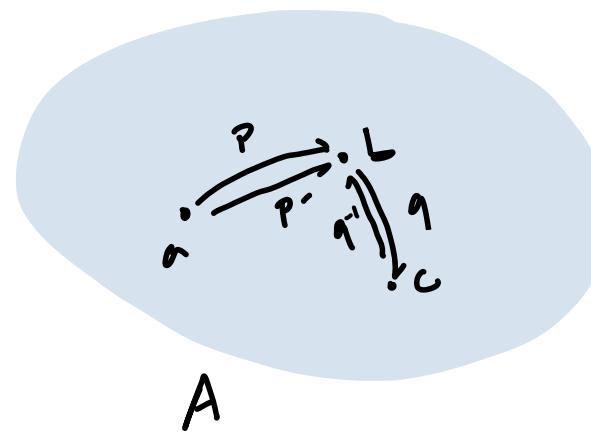
We can now think of types as collections of points (terms) connected by homotopies/paths (equalities).

We can:

- have multiple equalities of the same type (ex: $p, p' : a =_A b$)

The groupoidal behaviour of types

(The first homotopical phenomena)



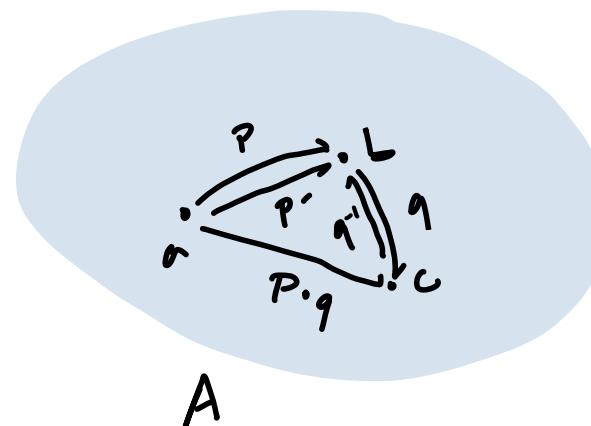
We can now think of types as collections of points (terms) connected by homotopies/paths (equalities).

We can:

- have multiple equalities of the same type (ex: $p, p': a =_A b$)
- take the inverse of an equality (if $q: b =_A c$, then $q^{-1}: c =_A b$)

The groupoidal behaviour of types

(The first homotopical phenomena)



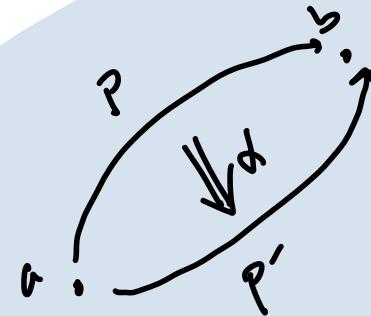
We can now think of types as collections of points (terms) connected by homotopies/paths (equalities).

We can:

- have multiple equalities of the same type (ex: $p, p': a \underset{A}{=} b$)
- take the inverse of an equality (if $q: b \underset{A}{=} c$, then $q^{-1}: c \underset{A}{=} b$)
- take composition of equalities (if $p: a \underset{A}{=} b$ and $q: b \underset{A}{=} c$, then $p \cdot q: a \underset{A}{=} c$)

The groupoidal behaviour of types

(The first homotopical phenomena)



A

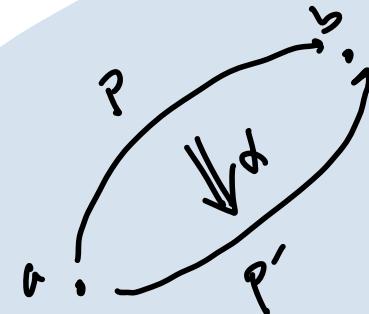
We can now think of types as collections of points (terms) connected by homotopies/paths (equalities).

We can:

- have multiple equalities of the same type (ex: $p, p': a \underset{A}{=} b$)
- take the inverse of an equality (if $q: b \underset{A}{=} c$, then $q^{-1}: c \underset{A}{=} b$)
- take composition of equalities (if $p: a \underset{A}{=} b$ and $q: b \underset{A}{=} c$, then $p \cdot q: a \underset{A}{=} c$)
- have equalities of equalities ($\alpha: p \underset{a \underset{A}{=} b}{=} p'$)

The groupoidal behaviour of types

(The first homotopical phenomena)



We can now think of types as collections of points (terms) connected by homotopies / paths (equalities).

We can:

- have multiple equalities of the same type (ex: $p, p': a \underset{A}{=} b$)
- take the inverse of an equality (if $q: b \underset{A}{=} c$, then $q^{-1}: c \underset{A}{=} b$)
- take composition of equalities (if $p: a \underset{A}{=} b$ and $q: b \underset{A}{=} c$, then $p \cdot q: a \underset{A}{=} c$)
- have equalities of equalities ($d: p \underset{a \underset{A}{=} b}{=} p'$)

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

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

$\prod_{b:B} E(b) \rightsquigarrow$

solutions of $\pi : \sum_{b:B} E(b) \rightarrow B$

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

$$\frac{a:A \vdash r_a : a =_A a}{\underline{a,b:A, p: a =_A b \vdash \text{ind}_=(r, a, b, p) : b =_A a}}$$

$$\lambda a, b, p. \text{ind}_=(r, a, b, p) : \prod_{a,b:A} a =_A b \rightarrow b =_A a.$$

= - elim

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

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

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

= - elim

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

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

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

? : $\prod_{a,b,c:A} a =_A b \longrightarrow b =_A c \longrightarrow a =_A c$

= - elim

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

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

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

$$a, b, c : A, p : a =_A b \vdash \quad ? \quad : b =_A c \longrightarrow a =_A c$$

$$? : \prod_{a,b,c:A} a =_A b \longrightarrow b =_A c \longrightarrow a =_A c$$

= - elim

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

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

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

$$\frac{c, a, b : A, p : a =_A b \vdash \quad ? \quad : b =_A c \longrightarrow a =_A c}{a, b, c : A, p : a =_A b \vdash \quad ? \quad : b =_A c \longrightarrow a =_A c}$$

$$? : \prod_{a,b,c:A} a =_A b \longrightarrow b =_A c \longrightarrow a =_A c$$

= - elim

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

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

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

$$\frac{\begin{array}{c} \ell, a:A \vdash ? : a =_A c \longrightarrow a =_A c \\ \hline c, a, b : A, p : a =_A b \vdash ? : b =_\ell c \longrightarrow a =_A c \end{array}}{a, b, c : A, p : a =_A b \vdash ? : b =_\ell c \longrightarrow a =_A c}$$

$$? : \prod_{a,b,c:A} a =_A b \longrightarrow b =_A c \longrightarrow a =_A c$$

= - elim

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

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

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

$$\frac{c, a : A \vdash \lambda x. x : a =_A c \longrightarrow a =_A c}{c, a, b : A, p : a =_A b \vdash ? : b =_c \longrightarrow a =_A c}$$

$$\frac{a, b, c : A, p : a =_A b \vdash ? : b =_c \longrightarrow a =_A c}{? : \prod_{a,b,c:A} a =_A b \longrightarrow b =_A c \longrightarrow a =_A c}$$

$$? : \prod_{a,b,c:A} a =_A b \longrightarrow b =_A c \longrightarrow a =_A c$$

= - elim

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

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

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

$$\frac{}{c, a : A \vdash \lambda x. x : a =_A c \longrightarrow a =_A c}$$

$$\frac{}{c, a, b : A, p : a =_A b \vdash \text{ind}_=(\lambda x. x, a, b, p) : b =_c c \longrightarrow a =_A c}$$

$$\frac{}{a, b, c : A, p : a =_A b \vdash ? : b =_A c \longrightarrow a =_A c}$$

$$\frac{? : \prod_{a,b,c:A} a =_A b \longrightarrow b =_A c \longrightarrow a =_A c}{? : \prod_{a,b,c:A} a =_A b \longrightarrow b =_A c \longrightarrow a =_A c}$$

= - elim

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

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

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

$$\frac{}{c, a : A \vdash \lambda x. x : a =_A c \longrightarrow a =_A c}$$

$$\frac{}{c, a, b : A, p : a =_A b \vdash \text{ind}_=(\lambda x. x, a, b, p) : b =_c c \longrightarrow a =_A c}$$

$$\frac{}{a, b, c : A, p : a =_A b \vdash \text{ind}_=(\lambda x. x, a, b, p) : b =_A c \longrightarrow a =_A c}$$

$$? : \prod_{a,b,c:A} a =_A b \longrightarrow b =_A c \longrightarrow a =_A c$$

= - elim

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

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

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

$$\frac{}{c, a : A \vdash \lambda x. x : a =_A c \longrightarrow a =_A c}$$

$$\frac{}{c, a, b : A, p : a =_A b \vdash \text{ind}_=(\lambda x. x, a, b, p) : b =_c c \longrightarrow a =_A c}$$

$$\frac{}{a, b, c : A, p : a =_A b \vdash \text{ind}_=(\lambda x. x, a, b, p) : b =_A c \longrightarrow a =_A c}$$

$$\frac{}{\lambda a, b, c, p. \text{ind}_=(\lambda x. x, a, b, p) : \prod_{a,b,c:A} a =_A b \longrightarrow b =_A c \longrightarrow a =_A c}$$

= - elim

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

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

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 $\text{tr}_p : E(b) \rightarrow E(b')$.

- This ensures that everything respects propositional equality. If we think of E as a predicate on B , then if $E(b)$ is true and $b =_B b'$, so is $E(b')$.
- This is part of a more sophisticated relationship between type theory and homotopy theory (Quillen model category theory). Transport says that $\pi : \sum_{b:B} E(b) \xrightarrow{\delta} B$ behaves like a fibration in a QMC.

Transport

$$\prod_{b,b':B} (b=b') \rightarrow E(b) \longrightarrow E(b')$$

= - elim

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

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

Transport

$$\prod_{b,b':B} (b = b') \rightarrow E(b) \longrightarrow E(b')$$

$$? : \prod_{b,b':B} (b = b') \rightarrow E(b) \longrightarrow E(b')$$

= - elim

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

Transport

$$\prod_{b,b':B} (b=b') \rightarrow E(b) \longrightarrow E(b')$$

$$b, b': B, p: b=b' \vdash ? : E(b) \longrightarrow E(b')$$

$$? : \prod_{b,b':B} (b=b') \rightarrow E(b) \longrightarrow E(b')$$

= - elim

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

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

Transport

$$\prod_{b,b':B} (b=b') \rightarrow E(b) \rightarrow E(b')$$

$$b:B \vdash ? : E(b) \rightarrow E(b)$$

$$b,b':B, p: b=b' \vdash ? : E(b) \rightarrow E(b')$$

$$? : \prod_{b,b':B} (b=b') \rightarrow E(b) \rightarrow E(b')$$

= - elim

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

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

Transport

$$\prod_{b,b':B} (b=b') \rightarrow E(b) \longrightarrow E(b')$$

$$b:B \vdash \lambda x.x: E(b) \rightarrow E(b)$$

$$b,b':B, p: b=b' \vdash ? : E(b) \longrightarrow E(b')$$

$$? : \prod_{b,b':B} (b=b') \rightarrow E(b) \longrightarrow E(b')$$

= - elim

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

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

Transport

$$\prod_{b,b':B} (b=b') \rightarrow E(b) \rightarrow E(b')$$

$$b:B \vdash \lambda x.x: E(b) \rightarrow E(b)$$

$$b,b':B, p: b=b' \vdash \text{ind}_=(\lambda x.x, b, b', p): E(b) \rightarrow E(b')$$

$$? : \prod_{b,b':B} (b=b') \rightarrow E(b) \rightarrow E(b')$$

= - elim

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

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

Transport

$$\prod_{b,b':B} (b=b') \rightarrow E(b) \rightarrow E(b')$$

$$b:B \vdash \lambda x.x: E(b) \rightarrow E(b)$$

$$b,b':B, p: b=b' \vdash \text{ind}_=(\lambda x.x, b, b', p): E(b) \rightarrow E(b')$$

$$\lambda b,b',p. \text{ind}_=(\lambda x.x, b, b', p): \prod_{b,b':B} (b=b') \rightarrow E(b) \rightarrow E(b')$$

= - elim

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

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

The homotopical content so far

- Types behave like spaces.
- However, UIP (the principle of uniqueness of identity proofs) is still consistent with what we have introduced thus far.

$$\text{UIP}(A) := \frac{\text{TT} \quad \text{TT}}{a,b:A \quad p,q:a=_A^b} \quad P = \frac{q}{a=_A^b}$$

- I.e., we still have an interpretation into sets.
- Only with higher inductive types or the univalence axiom honest homotopical content.
- I.e., we won't have an interpretation into sets.