Outline:

Last time: inductive types – identity type $\Big\}$ §3-4 Rijke

This time: identity type $\Big\}$ §5

→ homotopy

Why do we need the identity type?

(If we're not interested in homotopy.)

We already have a notion of equality:

<u>judgmental equality</u> $\doteq$

(The identity type is called <u>propositional equality</u> $=$.)

<u>Logical interpretation</u>: propositions are types / proofs are terms.

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

Why do we need the identity type actually?

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

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

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

Why do we need the identity type actually?

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

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

... but not all the equalities we want!

Ex. One cannot prove $\mathrm{add}(0,x) \doteq x$.

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

$\mathbb{N}$-elim (roughly):

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

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
- $\mathbb{N}$
- $\emptyset$
- $\mathbb{1}$

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

- The universe type (next lecture) internalizes the judgment of the form

  A type

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

Identity type $=$

$=$-form

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

$=$-intro

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

$=$-elim

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

$=$-comp

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

Identity type =

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

$=$-form

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

$=$-intro

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

$=$-elim

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

$=$-comp

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

Identity type $=$

$=$-form

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

$=$-intro

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

$=$-elim

$$\frac{\begin{array}{r} x : A, y : A, z : x =_A y \vdash D(x,y,z)\ \mathrm{type} \\ x : A \vdash d : D(x,x,r_x) \end{array}}{x : A, y : A, z : x =_A y \vdash \mathrm{ind}_= (d, x, y, z) : D(x,y,z)\ \mathrm{type}}$$

$=$-comp

$$\frac{\begin{array}{r} x : A, y : A, z : x =_A y \vdash D(x,y,z)\ \mathrm{type} \\ x : A \vdash d : D(x,x,r_x) \end{array}}{x : A \vdash \mathrm{ind}_= (d, x, x, r_x) \doteq d : D(x,x,r_x)}$$

Identity type = with based path induction

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

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

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

$=^b$ - elim
$$\frac{\begin{array}{c} a : A \\ x : A,\, z : a =_A x \vdash D(x, z) \text{ type} \\ \vdash d : D(a, r_a) \end{array}}{x : A, z : a =_A x \vdash \mathrm{ind}^b_=(d, x, z) : D(x, z) \text{ type}}$$

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

$=^b$ - comp
$$\frac{\begin{array}{c} a : A \\ x : A, z : x =_A y \vdash D(x, z) \text{ type} \\ \vdash d : D(a, r_a) \end{array}}{\vdash \mathrm{ind}^b_=(d, a, r_a) \doteq d : D(a, r_a)}$$

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

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.

Functoriality

Functions act on paths.

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

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

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

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

Functoriality:

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

= - elim

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

Functoriality:

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

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

$=$ - elim

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

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

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

= - elim

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

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

$$\dfrac{\dfrac{f:A\to B,\ a \vdash \ ? \ : fa =_B fa}{f:A\to B,\ a,a':A,\ p: a =_A a' \vdash \quad ? \quad : fa =_B fa'}}{? \ : \prod_{f:A\to B} \prod_{a,a':A} a =_A a' \to fa =_B fa'}$$

= - elim

$$\dfrac{\begin{array}{r} x:A, y:A, z: x =_A y \vdash D(x,y,z) \text{ type} \\ x:A \vdash d: D(x,x,r_x) \end{array}}{x:A, y:A, z: x =_A y \vdash \mathrm{ind}_=(d,x,y,z) : D(x,y,z) \text{ type}}$$

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

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

$=$ - elim

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

Functoriality:

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

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

= - elim

$$\dfrac{\begin{array}{r} x:A, y:A, z: x =_A y \vdash D(x,y,z) \text{ type} \\ x:A \vdash d : D(x,x,r_x) \end{array}}{x:A, y:A, z: x =_A y \vdash \mathrm{ind}_=(d,x,y,z) : D(x,y,z) \text{ type}}$$

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

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

= - elim

$$\cfrac{\begin{array}{r} x:A, y:A, z: x =_A y \vdash D(x,y,z) \text{ type} \\ x:A \vdash d : D(x,x,r_x) \end{array}}{x:A, y:A, z: x =_A y \vdash \mathrm{ind}_=(d,x,y,z) : D(x,y,z) \text{ type}}$$

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

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

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

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

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

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

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

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

Use: $add(n,0) \doteq n$

$add(n, sm) \doteq s\, add(n,m)$

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

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

Use: $add(n,0) \doteq n$

$add(n,sm) \doteq s\, add(n,m)$

$$\dfrac{\dfrac{? : add(0,0) = 0 \qquad n:\mathbb{N},\ p : add(0,n) = n \vdash\ ?\ : add(0,sn) = sn}{n:\mathbb{N} \vdash\ ?\ : add(0,n) = n}}{?\ : \prod_{n:\mathbb{N}} add(0,n) = n}$$

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

Use: $add(n,0) \doteq n$

$add(n,sm) \doteq s\, add(n,m)$

$$\dfrac{r_0 : add(0,0) = 0 \qquad n:\mathbb{N},\ p: add(0,n)=n \vdash ap_s p : add(0,sn) = sn}{\dfrac{n:\mathbb{N} \vdash \quad ? \quad : add(0,n) = n}{? \quad : \prod_{n:\mathbb{N}} add(0,n) = n}}$$

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

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

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

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

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

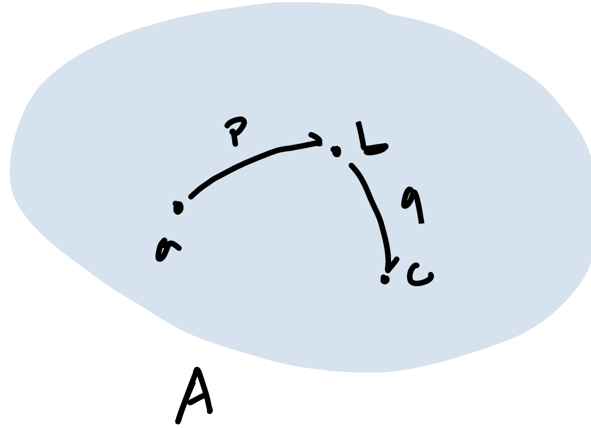
Use: $\mathrm{add}(n,0) \doteq n$

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

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

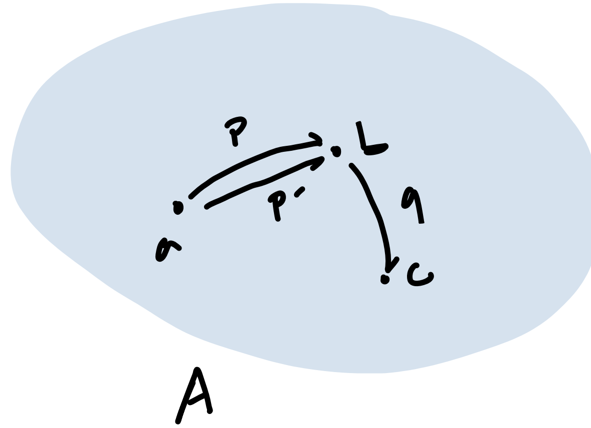
The groupoidal behaviour of types

(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)

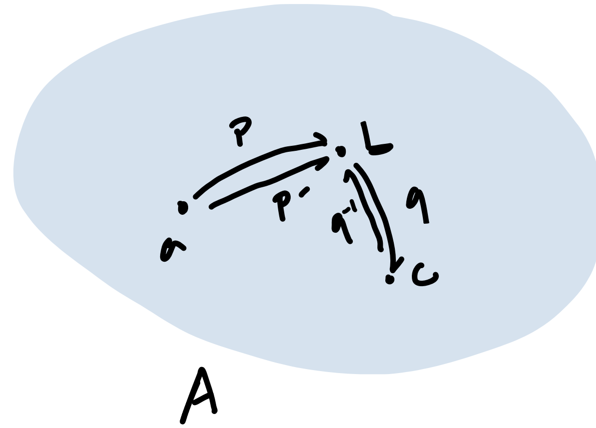
We can now think of types as collections of points (terms) connected by homotopies/paths (equalities).

We can:

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

The groupoidal behaviour of types

(The first homotopical phenomena)

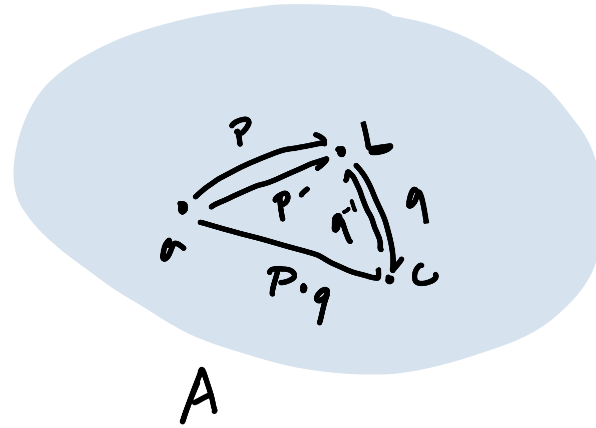


We can now think of types as collections of points (terms) connected by homotopies/paths (equalities).

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)

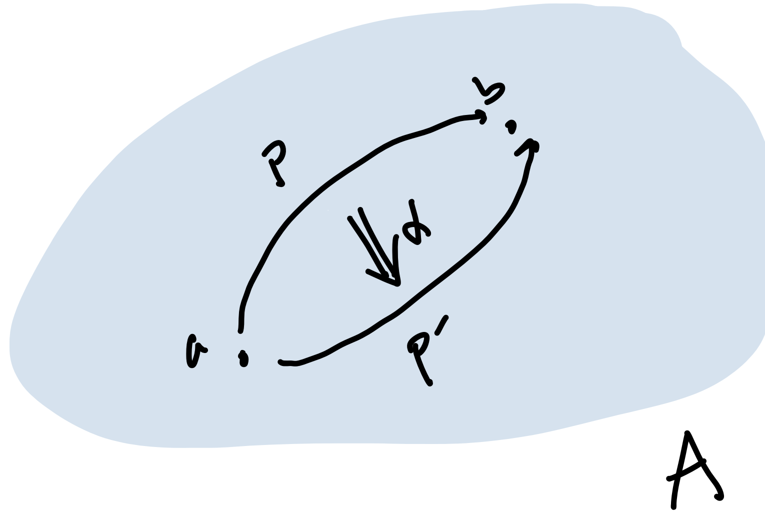
We can now think of types as collections of points (terms) connected by homotopies/paths (equalities).

We can:

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

The groupoidal behaviour of types

(The first homotopical phenomena)



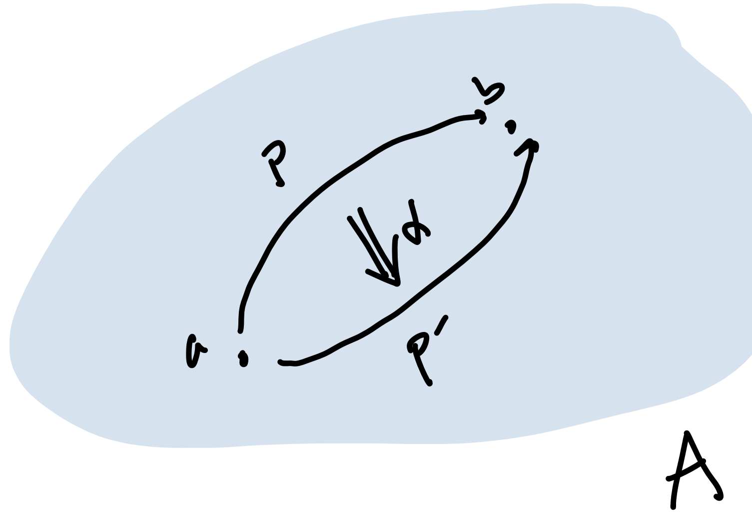
We can now think of types as collections of points (terms) connected by homotopies/paths (equalities).

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 $q^{-1} : c =_A b$)
- take composition of equalities (if $p : a =_A b$ and $q : b =_A c$, then $p \cdot q : a =_A c$)
- have equalities of equalities ($\alpha : p =_{a =_A b} p'$)

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

This is how <u>homotopies</u> in <u>spaces</u> 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
$\pi : \sum_{b:B} E(b) \to B$	$\rightsquigarrow$		Kan fibrations
$\prod_{b:B} E(b)$	$\rightsquigarrow$		sections of $\pi : \sum_{b:B} E(b) \to B$

__Inverse of equalities__: $\prod_{a,b:A} a =_A b \to b =_A a.$

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

$=$ - elim

$$\dfrac{\begin{array}{c} x:A, y:A, z: x =_A y \vdash D(x,y,z) \text{ type} \\ x:A \vdash d : D(x,x,r_x) \end{array}}{x:A, y:A, z: x =_A y \vdash \mathrm{ind}_=(d,x,y,z) : D(x,y,z) \text{ type}}$$

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

= - elim

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

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

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

= - elim

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

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

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

= - elim

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

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

$$\dfrac{\dfrac{c, a, b : A,\ p : a =_A b \vdash\ ?\ : b =_A c \to a =_A c}{a, b, c : A,\ p : a =_A b \vdash\ ?\ : b =_A c \to a =_A c}}{?\ : \prod_{a,b,c:A} a =_A b \to b =_A c \to a =_A c}$$

= - elim

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

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

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

= - elim

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

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

$$\cfrac{\cfrac{\cfrac{c, a : A \vdash \lambda x . x : a =_A c \longrightarrow a =_A c}{c, a, b : A,\ p : a =_A b \vdash \quad ? \quad : b =_A c \longrightarrow a =_A c}}{a, b, c : A,\ p : a =_A b \vdash \quad ? \quad : b =_A c \longrightarrow a =_A c}}{? \ : \prod_{a,b,c:A} a =_A b \longrightarrow b =_A c \longrightarrow a =_A c}$$

= - elim

$$\cfrac{\begin{array}{r} x : A, y : A, z : x =_A y \vdash D(x,y,z) \text{ type} \\ x : A \vdash d : D(x,x,r_x) \end{array}}{x : A, y : A, z : x =_A \ \vdash \mathrm{ind}_=(d,x,y,z) : D(x,y,z) \text{ type}}$$

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

$$
\dfrac{\dfrac{\dfrac{c, a : A \vdash \lambda x.x : a =_A c \longrightarrow a =_A c}{c, a, b : A,\ p : a =_A b \vdash \mathrm{ind}_=(\lambda x.x, a, b, p) : b =_A c \longrightarrow a =_A c}}{a, b, c : A,\ p : a =_A b \vdash \quad ? \quad : b =_A c \longrightarrow a =_A c}}{? \ : \prod_{a,b,c:A} a =_A b \longrightarrow b =_A c \longrightarrow a =_A c}
$$

$=$ - elim

$$
\dfrac{\begin{array}{r} x:A, y:A, z: x =_A y \vdash D(x,y,z) \text{ type} \\ x:A \vdash d : D(x,x,r_x) \end{array}}{x:A, y:A, z: x =_A \ \vdash \mathrm{ind}_=(d,x,y,z) : D(x,y,z) \text{ type}}
$$

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

$$\begin{array}{c}
c, a : A \vdash \lambda x . x : a =_A c \longrightarrow a =_A c \\
\hline
c, a, b : A, p : a =_A b \vdash \mathrm{ind}_=(\lambda x . x, a, b, p) : b =_A c \longrightarrow a =_A c \\
\hline
a, b, c : A, p : a =_A b \vdash \mathrm{ind}_=(\lambda x . x, a, b, p) : b =_A c \longrightarrow a =_A c \\
\hline
? \ : \prod_{a,b,c:A} a =_A b \longrightarrow b =_A c \longrightarrow a =_A c
\end{array}$$

= - elim

$$\begin{array}{c}
x : A, y : A, z : x =_A y \vdash D(x,y,z) \text{ type} \\
x : A \vdash d : D(x,x,r_x) \\
\hline
x : A, y : A, z : x =_A \vdash \mathrm{ind}_=(d,x,y,z) : D(x,y,z) \text{ type}
\end{array}$$

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

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

= - elim

$$\dfrac{\begin{array}{c} x : A, y : A, z : x =_A y \vdash D(x, y, z)\ \text{type} \\ x : A \vdash d : D(x, x, r_x) \end{array}}{x : A, y : A, z : x =_A \quad \vdash \mathrm{ind}_=(d, x, y, z) : D(x, y, z)\ \text{type}}$$

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 $tr_B : E(b) \to E(b')$.

- 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 $\pi : \sum_{b:B} E(b) \longrightarrow B$ behaves like a fibration in a QMC.

Transport

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

$=$ - elim

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

Transport

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

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

$=$ - elim

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

Transport

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

$$\frac{b, b' : B,\ p : b = b' \vdash \quad ? \quad : E(b) \longrightarrow E(b')}{? \ : \prod_{b,b':B} (b = b') \longrightarrow E(b) \longrightarrow E(b')}$$

= - elim

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

Transport

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

$$\cfrac{\cfrac{b : B \vdash \; ? \; : E(b) \to E(b)}{b, b' : B,\ p : b = b' \vdash \quad ? \quad : E(b) \to E(b')}}{? : \prod_{b,b':B} (b = b') \longrightarrow E(b) \longrightarrow E(b')}$$

= - elim

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

Transport

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

$$\dfrac{\dfrac{b:B \vdash \lambda x.x : E(b) \longrightarrow E(b)}{b,b':B,\ p: b=b' \vdash \quad ? \quad : E(b) \longrightarrow E(b')}}{? \ : \prod_{b,b':B} (b=b') \longrightarrow E(b) \longrightarrow E(b')}$$

$=$ - elim

$$\dfrac{\begin{array}{r} x:A, y:A, z: x=_A y \vdash D(x,y,z) \text{ type} \\ x:A \vdash d: D(x,x,r_x) \end{array}}{x:A, y:A, z: x=_A \ \vdash \mathrm{ind}_=(d,x,y,z) : D(x,y,z) \text{ type}}$$

Transport

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

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

= - elim

$$\frac{\begin{array}{r} x:A, y:A, z : x =_A y \vdash D(x,y,z)\ \mathrm{type} \\ x:A \vdash d : D(x,x,r_x) \end{array}}{x:A, y:A, z : x =_A \vdash \mathrm{ind}_=(d, x, y, z) : D(x,y,z)\ \mathrm{type}}$$

Transport

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

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

= - elim

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

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.

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

- 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.

(CODA)

Uniqueness of canonical terms in Σ-types

We have $\pi : \sum_{b:B} E(b) \longrightarrow B$ and we can construct

$$\rho : \prod_{x : \sum_{b:B} E(b)} E(\pi x)$$

Then we can show

$$\prod_{x : \sum_{b:B} E(b)} x = \mathrm{pair}(\pi x, \rho x)$$

This is a propositional η-rule for Σ types.

There is a similar propositional η-rule for Id types if you set it up correctly (see exercises).