The Hypercubic Manifold in Homotopy Type Theory

Dylan Laird

04/09/2023

Table of contents

Table of contents

Type Theory

We manipulate types A, B and terms $a : A$ of some certain type.

Type Theory

We manipulate types A, B and terms $a : A$ of some certain type.

The inductive type of natural integers

- $0 : \mathbb{N}$
- $\mathrm{suc} : \mathbb{N} \to \mathbb{N}$

Type Theory

We manipulate types A, B and terms $a : A$ of some certain type.

The inductive type of natural integers

- $0 : \mathbb{N}$
- $\mathrm{suc} : \mathbb{N} \to \mathbb{N}$

Equality

We distinguish between **definitional equality** $\equiv$ used when defining objects and **propositional equality** which is a type theoretical concept.

Type Theory

We manipulate types A, B and terms $a : A$ of some certain type.

The inductive type of natural integers

- $0 : \mathbb{N}$
- $\mathrm{suc} : \mathbb{N} \to \mathbb{N}$

Equality

We distinguish between **definitional equality** $\equiv$ used when defining objects and **propositional equality** which is a type theoretical concept.

Propositional Equality

Given a type A and terms $a, b : A$ there is a type $a =_A b$. We say that elements a and b are (propositionally) equal when $a =_A b$ is inhabited.

Some constructions on types

Function types

Given types A, B one has a type $A \to B$ of functions from A to B.

Some constructions on types

Function types

Given types A, B one has a type $A \to B$ of functions from A to B.

Product type

Given types A, B one has a product type $A \times B$.

Some constructions on types

Function types

Given types A, B one has a type $A \to B$ of functions from A to B.

Product type

Given types A, B one has a product type $A \times B$.

Coproduct type

Give types A, B one has a coproduct type $A + B$ given by the constructors :

- $\mathrm{inl} : A \to A + B$
- $\mathrm{inr} : B \to A + B$

Manipulating types

Introduction rules

They encapsulate how to build an element of a certain type.

- To build an element of $f : A \to B$ one needs an expression $\phi(x)$ such that $a : A \vdash \phi(a) : A$ and to set $f :\equiv \lambda x.\phi(x)$
- To build an element of $A \times B$ one needs to take elements $a : A$ and $b : B$ to form $(a, b) : A \times B$

Manipulating types

Introduction rules

They encapsulate how to build an element of a certain type.

- To build an element of $f : A \to B$ one needs an expression $\phi(x)$ such that $a : A \vdash \phi(a) : A$ and to set $f :\equiv \lambda x.\phi(x)$
- To build an element of $A \times B$ one needs to take elements $a : A$ and $b : B$ to form $(a, b) : A \times B$

Induction principles

They encapsulate how to build **dependent** functions from a source type A.

- To build a function of type $f : \prod_{x:A\times B} P(x)$ one only needs to give its value on pairs (a, b).

A working example : product types (1)

Projections

We can define projections $\mathrm{pr}_1 : A \times B \to A$ and $\mathrm{pr}_2 : A \times B \to B$ by the **induction** principle for product types by setting $\mathrm{pr}_1((a, b)) :\equiv a$ and $\mathrm{pr}_2((a, b)) :\equiv b$.

A working example : product types (1)

Projections

We can define projections $\mathrm{pr}_1 : A \times B \to A$ and $\mathrm{pr}_2 : A \times B \to B$ by the **induction** principle for product types by setting $\mathrm{pr}_1((a, b)) :\equiv a$ and $\mathrm{pr}_2((a, b)) :\equiv b$.

Propositional uniqueness for product types

We have a type theoretic statement that expresses that the type $A \times B$ is the type of "pairs of elements of A and B" :

$$\mathrm{uniq}_{A\times B} : \prod_{x:A\times B} \Big(x =_{A\times B} (\mathrm{pr}_1(x), \mathrm{pr}_2(x)) \Big)$$

A working example : product types (2)

Sketch of proof :

- By induction, we only need to build an element :

$$\mathrm{uniq}_{A\times B}((a,b)) : (a,b) = (\mathrm{pr}_1(a,b), \mathrm{pr}_2(a,b))$$

A working example : product types (2)

Sketch of proof :

- By induction, we only need to build an element :

$$\mathrm{uniq}_{A\times B}((a,b)) : (a,b) = (\mathrm{pr}_1(a,b), \mathrm{pr}_2(a,b))$$

- By the definition of the projections, the goal type reduces to:

$$(a,b) =_{A\times B} (a,b)$$

A working example : product types (2)

Sketch of proof :

- By induction, we only need to build an element :

$$\mathrm{uniq}_{A\times B}((a,b)) : (a,b) = (\mathrm{pr}_1(a,b), \mathrm{pr}_2(a,b))$$

- By the definition of the projections, the goal type reduces to:

$$(a,b) =_{A\times B} (a,b)$$

- We now have an element refl : $(a,b) =_{A\times B} (a,b)$ $\square$