Synthetic Homotopy Theory TD1: Inductive types

January 21, 2020

An inductive type is defined by a list of constructors, allowing you to build inahbitants of the type, and an elimination principle allowing you to build maps out of the type (or equivalently prove things for all inhabitants of the type). In this first section we will study identity types of some examples of inductive types. In class we saw how to compute the identity types in product types using the *encode-decode method.* In summary:

1. We define a family $\mathrm{Eq}_{A\times B} : A\times B \to A\times B \to \mathcal{U}$.
2. We define functions in:
$$\textbf{encode} : (z, z' : \mathrm{Eq}_{A\times B}(z,z')) \to z = z' \to \mathrm{Eq}_{A\times B}(z,z')$$
and
$$\textbf{decode} : (z, z' : \mathrm{Eq}_{A\times B}(z,z')) \to \mathrm{Eq}_{A\times B}(z,z') \to z = z'$$
3. We show that they are inverse to each other.
4. From this we conclude that:
$$(x,y) =_{A\times B} (x',y') \simeq (x =_A x') \times (y =_B y')$$

Now we try to apply this method to other types.

Exercise 1 The unit type

We define $\mathrm{Eq}_{\mathbf{1}} : \mathbf{1} \to \mathbf{1} \to \mathcal{U}$ by:

$$\mathrm{Eq}_{\mathbf{1}}(*,*) :\equiv \mathbf{1}$$

Question 1 Prove that for any $A : \mathcal{U}$ we have $(A\times \mathbf{1}) \simeq A$ and that $(A \to \mathbf{1}) \simeq \mathbf{1}$

Question 2 Define a function:

$$\textbf{encode} : (x, y : \mathbf{1}) \to x = y \to \mathrm{Eq}_{\mathbf{1}}(x,y)$$

(Hint: start by path induction).

Question 3 Define a function:

$$\textbf{decode} : (x, y : \mathbf{1}) \to \mathrm{Eq}_{\mathbf{1}}(x,y) \to x = y$$

(Hint: start by induction on x and y).

Question 4 Prove that **encode** and **decode** are inverse to each other. Conclude that:

$$(* = *) \simeq \mathbf{1}$$

From this conclude that:

$$(x = y) \simeq \mathbf{1}$$

for any $x, y : \mathbf{1}$.

Exercise 2 The empty type

Question 1 Prove that for all $A : \mathcal{U}$ we have $(A \times \mathbf{0}) \simeq A$ and $(\mathbf{0} \to A) \simeq \mathbf{1}$.

Question 2 Prove that for all $x, y : \mathbf{0}$, we have $x = y$.

Question 3 Recall that $x \neq y$ is defined as $x = y \to \mathbf{0}$. Prove that for all $x, y : \mathbf{0}$, we have $x \neq y$.

Question 4 Prove that for all $x, y : \mathbf{0}$, anything holds for $x =_{\mathbf{0}} y$.

Question 5 Let A be a type. Show that any map in $A \to \mathbf{0}$ is an equivalence.

Exercise 3 The type of booleans

We define $\mathrm{Eq}_{\mathbf{2}} : \mathbf{2} \to \mathbf{2} \to \mathcal{U}$ by:

$$\begin{aligned}
\mathrm{Eq}_{\mathbf{2}}(0,0) &:\equiv \mathbf{1}\\
\mathrm{Eq}_{\mathbf{2}}(0,1) &:\equiv \mathbf{0}\\
\mathrm{Eq}_{\mathbf{2}}(1,0) &:\equiv \mathbf{0}\\
\mathrm{Eq}_{\mathbf{2}}(1,1) &:\equiv \mathbf{1}
\end{aligned}$$

You should convince yourself that these are reasonable expectation for equalities in a two-point set.

Question 1 Define a function:

$$\textbf{encode} : (x, y : \mathbf{2}) \to x = y \to \mathrm{Eq}_{\mathbf{2}}(x,y)$$

(Hint: start by path induction).

Question 2 Define a function:

$$\mathbf{decode} : (x, y : \mathbf{2}) \to \mathrm{Eq}_{\mathbf{2}}(x, y) \to x = y$$

(Hint: start by induction on x and y).

Question 3 Prove that **encode** and **decode** are inverse to each other. Conclude that $0 \neq 1$.

Question 4 (Optional) Can you find a shorter proof that $0 \neq 1$?

Exercise 4 Sum types

We define:

$$\mathrm{Eq}_{A+B} : A + B \to A + B \to \mathcal{U}$$

by:

$$\begin{aligned}
\mathrm{Eq}_{A+B}(\mathbf{inc}_l(x), \mathbf{inc}_l(x')) &:\equiv x =_A x' \\
\mathrm{Eq}_{A+B}(\mathbf{inc}_l(x), \mathbf{inc}_r(y')) &:\equiv \mathbf{0} \\
\mathrm{Eq}_{A+B}(\mathbf{inc}_r(y), \mathbf{inc}_l(x')) &:\equiv \mathbf{0} \\
\mathrm{Eq}_{A+B}(\mathbf{inc}_r(y), \mathbf{inc}_l(y')) &:\equiv y =_B y'
\end{aligned}$$

You should convince yourself that these are reasonable expectation for equalities in a two-point set.

Question 1 Assume given $A, B, C : \mathcal{U}$, prove that $(A + B \to C) \simeq (A \to C) \times (B \to C)$

Question 2 Prove that for any $A : \mathcal{U}$ we have $(A + \mathbf{0}) \simeq A$.

Question 3 Prove that $\mathbf{1} + \mathbf{1} \simeq \mathbf{2}$.

Question 4 Using the same method as in exercises 1 and 3, prove that for all $z, z' : A + B$ we have:

$$(z = z') \simeq \mathrm{Eq}_{A+B}(z, z')$$

Exercise 5 The natural numbers

We define $\mathrm{Eq}_{\mathbb{N}} : \mathbb{N} \to \mathbb{N} \to \mathcal{U}$ by:

$$\begin{aligned}
\mathrm{Eq}_{\mathbb{N}}(0, 0) &:\equiv \mathbf{1} \\
\mathrm{Eq}_{\mathbb{N}}(0, \mathbf{s}(n)) &:\equiv \mathbf{0} \\
\mathrm{Eq}_{\mathbb{N}}(\mathbf{s}(m), 0) &:\equiv \mathbf{0} \\
\mathrm{Eq}_{\mathbb{N}}(\mathbf{s}(m), \mathbf{s}(n)) &:\equiv \mathrm{Eq}_{\mathbb{N}}(m, n)
\end{aligned}$$

Note that we used $\mathrm{Eq}_{\mathbb{N}}$ in the definition of $\mathrm{Eq}_{\mathbb{N}}$ which is okay because $\mathbb{N}$ is recursive.

Question 1 Using the same method as in exercises 1, 3 and 4 prove that for all $m, n : \mathbb{N}$ we have:

$$(m = n) \simeq \mathrm{Eq}_{\mathbb{N}}(m, n)$$

Question 2 Conclude that $0 \neq 1$, and that for any $m, n : \mathbb{N}$ we have:

$$(m = n) \simeq (\mathbf{s}(m) = \mathbf{s}(n))$$

Exercise 6 Univalence contradicts unicity of identity proofs

Recall that the unicity of identity proofs (UIP for short) is the principle that for all $p, q : x =_A y$ we have $p = q$. We will use the type of booleans $\mathbf{2}$ to prove that univalence contradicts UIP.

Question 1 Define an equivalence $\mathbf{swap} : \mathbf{2} \simeq \mathbf{2}$ swapping 0 and 1.

Question 2 Show that $\mathbf{swap} \neq \mathbf{id}$ in $\mathbf{2} \simeq \mathbf{2}$.

Question 3 Conclude that we have two distinct elements in $\mathbf{2} =_{\mathcal{U}} \mathbf{2}$ using univalence.

* Exercise 7 Quasi-equivalences

Recall that for $f : A \to B$ we defined $\mathrm{isQuasiEquiv}(f)$ by:

$$\mathrm{isQuasiEquiv}(f) :\equiv (g : B \to A) \times (f \circ g \sim \mathbf{id}_B) \times (g \circ f \sim \mathbf{id}_A)$$

Question 1 Assume we want to prove something for all types $A, B : \mathcal{U}$ and all equivalences $\epsilon : A \simeq B$. Using univalence and path induction show that it is enough to prove what we want for the equivalences $\mathbf{id}_A$ for any $A : \mathcal{U}$.

Question 2 Using the previous question, prove that for all equivalences $\epsilon : A \simeq B$ we have:

$$\mathrm{isQuasiEquiv}(\epsilon) \simeq (a : A) \to a = a$$

* Exercise 8 Types of lists, and more

We assume that you are familiar with the inductive type $\mathbf{List}_A$ of lists of elements in A. We denote by $\emptyset$ the empty list and $a :: l$ the list with head a and tail l.

Question 1 Using the encode-decode method, prove that:

$$(a :: l =_{\mathbf{List}_A} a' :: l') \simeq (a =_A a) \times (l =_{\mathbf{List}_A} l')$$

Question 2 If you happen to know other inductive types (e.g. W-types), what can you prove about their identity types using the encode-decode method ?