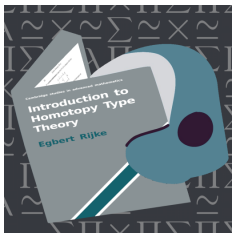

Worksheet 3 (Solved)

HoTTEST Summer School 2022

The HoTTEST TAs

11 July 2022

1 $(\star)$

Let A be a type, and $x, y, z : A$. Show that path inversion distributes over concatention, i.e., construct a term of the following type:

$$\Pi_{p:x=y}\Pi_{q:y=z}(p \cdot q)^{-1} = q^{-1} \cdot p^{-1}$$

By path induction on p, then q, it suffices to give a term of type

$$(\mathsf{refl}_x \cdot \mathsf{refl}_x)^{-1} = \mathsf{refl}_x^{-1} \cdot \mathsf{refl}_x^{-1}$$

Both sides are definitionally refl_x, so we can give the term $\mathsf{refl}_{\mathsf{refl}_x}$.

2 $(\star\star)$

Let A and B be types. We can define the product type as $A \times B := \Sigma_A B$, where B is considered a constant type family over A. The resulting elimination principle is:

$$\frac{\Gamma, z : A \times B \vdash D(z)\ \text{type} \quad \Gamma, a : A, b : B \vdash d : D(a, b)}{\Gamma, z : A \times B \vdash \mathsf{ind}_\times(d, z) : D(z)}\ \times\text{−Elim}$$

Define the two projections $\mathsf{pr}_1 : A \times B \to A$ and $\mathsf{pr}_2 : A \times B \to B$ using the elimination principle.

For pr_1 we let D be the constant type family at A, and we let d be a:

$$\frac{\Gamma, z : A \times B \vdash A\ \textbf{type} \quad \Gamma, a : A, b : B \vdash a : A}{\Gamma, z : A \times B \vdash \mathsf{pr}_1(z) : A}\ \times\textbf{−Elim}$$

For pr_2, let D be the constant type family at B and let d be b above.

Now use the elimination principle to give a term of the following type:

$$\Pi_{z:A\times B} z = (\mathsf{pr}_1 z, \mathsf{pr}_2 z)$$

The type family we're considering is $z : A \times B \vdash z = (\mathsf{pr}_1 z, \mathsf{pr}_2 z)$ type. In order to use the elimination principle, we need a term d as follows:

$$a : A, b : B \vdash d : (a, b) = (\mathsf{pr}_1(a, b), \mathsf{pr}_2(a, b))$$

By our definition of the projection maps, we have definitional equalities $\mathsf{pr}_1(a, b) \doteq a$ and $\mathsf{pr}_2(a, b) \doteq b$. So we can take $d \doteq \mathsf{refl}_{(a,b)}$ above. The elimination principle then gives us

$$z : A \times B \dashv \mathsf{ind}_\times(d, z) : z = (\mathsf{pr}_1 z, \mathsf{pr}_2 z)$$

which by abstraction over z gives the desired term.

3 $(\star)$

Let A and B be types. Give an informal construction of a term of the following type:

$$\Pi_{a,a':A}\Pi_{b,b':B}(a =_A a') \times (b =_B b') \to \big((a, b) =_{A\times B} (a', b')\big)$$

Let $a, a' : A$ and $b, b' : B$. By the elimination principle for product types, we may consider two paths $p : a = a'$ and $q : b = b'$. We need to construct a term of type $(a, b) =_{A\times B} (a', b')$. By path induction on p and q, it suffices to give a term of type $(a, b) =_{A\times B} (a, b)$. For this term, take $\mathsf{refl}_{(a,b)}$.

(We will later see that this map is an *equivalence*, and this map *characterises* paths in product types as pairs of paths between the components.)

4 $(\star\star)$

Let A be a type and $a : A$. Show that refl_a is unique among paths starting at a but with the other endpoint free. That is, for any $z : \Sigma_{x:A}(a = x)$ construct a term

$$(a, \mathsf{refl}_a) =_{\Sigma_{x:A}(a=x)} z$$

(We emphasize that this does not mean that refl_a is unique as a loop at a!)

By the elimination principle of Σ-types, we may assume $z \doteq (x, p)$ with $x : A$ and $p : a = x$. Then, by path induction on p it suffices to give a term of type $(a, \mathsf{refl}_a) = (a, \mathsf{refl}_a)$. Take $\mathsf{refl}_{(a,\mathsf{refl}_a)}$ for this term.

(This is Prop. 5.5.1 in Egbert's book.)

5 $(\star\star\star)$

Let A be a type. In the third lecture, a function concat of the following type was constructed:

$$\mathsf{concat} : \Pi_{x,y,z:A}(x = y) \to ((y = z) \to (x = z))$$

Call the above for concat_1. Define two different terms concat_2 and concat_3 of the same type.

We start by recalling the definition of concat_1.
(1) Let $x, y : A$. We can interchange the parameters $z : A$ and $p : x = y$ since they are independent (see exercise 5 from worksheet 1). By path induction on $p : x = y$ we may assume that $x \doteq y$, thus need to construct a term of type $\Pi_{z:A}(x = z) \to (x = z)$. By path induction on $q : (x = z)$, it suffices to give a term of type $(x = x)$. Let refl_x be this term. This defines $\mathsf{concat}_1(p, q)$, and $\mathsf{concat}_1(\mathsf{refl}_x, \mathsf{refl}_x) \doteq \mathsf{refl}_x$.
From now on, we implicitly interchange independent variables as needed.
(2) Let $x, y : A$ and $p : x = y$. We need to give a term of type $\Pi_{z:A}(y = z) \to (x = z)$. By path induction on p, we can assume $x \doteq y$, in which case $\mathsf{id}_{(x=z)}$ is a term of that type. This defines $\mathsf{concat}_2(p, q)$, and we have that $\mathsf{concat}_2(\mathsf{refl}, q) \doteq \mathsf{id}(q) \doteq q$.
(3) Define concat_3 similarly by inducting on q, but not p. This defines $\mathsf{concat}_3(p, q)$ such that $\mathsf{concat}_3(p, \mathsf{refl}) \doteq p$.

Choose your favorite numbers $i, j \in \{1, 2, 3\}$ $(i \neq j)$. Give a term of the following type:

$$\Pi_{x,y,z:A}\Pi_{p:x=y}\Pi_{q:y=z}\mathsf{concat}_i(p, q) = \mathsf{concat}_j(p, q)$$

Let's pick $i \doteq 2$ and $j \doteq 3$. By path induction on $p : (x = y)$, we need to give a term of type

$$\Pi_{q:(x=z)}\mathsf{concat}_2(\mathsf{refl}_x, q) = \mathsf{concat}_3(\mathsf{refl}_x, q)$$

Note that by definition, we have $\mathsf{concat}_2(\mathsf{refl}_x, q) \doteq q$. By path induction on q, it suffices to give a term of type

$$\mathsf{refl}_x = \mathsf{concat}_3(\mathsf{refl}_x, \mathsf{refl}_x)$$

The right-hand side is definitionally refl_x, so $\mathsf{refl}_{\mathsf{refl}_x}$ is a term of this type.

6 $(\star\star\star)$

Let A be a type with an element $a : A$. Can you construct a term of the following type? If not, what goes wrong?

$$\Pi_{p:(a=_A a)} p =_{(a=_A a)} \mathsf{refl}_a$$

We cannot apply path induction since both endpoints of p are fixed. Later in the course, we will be able to prove that a term of this type does not exist. We can begin to understand the intuition via the interpretation of "types as spaces" discussed in lecture 3. Under this interpretation, this type says that "all loops at $a : A$ are equal to refl_a." But there are spaces with non-trivial loops, such as the circle.