Homework 4: Second Identity Crisis

15-819 Homotopy Type Theory

Out: 4/Nov/13
Due: 18/Nov/13

1 Application Programming Interface

Assume $\Gamma \vdash f : A \to B$ and $\Gamma \vdash g : B \to C$. Here are some useful properties of ap.

$$\Gamma \vdash \underline{\quad\quad} : \prod_{m:A} \mathtt{ap}_f(\mathtt{refl}_A(m)) = \mathtt{refl}_B(f\,m)$$

$$\Gamma \vdash \underline{\quad\quad} : \prod_{m,n:A} \prod_{p:m=n} \mathtt{ap}_f(p^{-1}) = \mathtt{ap}_f(p)^{-1}$$

$$\Gamma \vdash \mathtt{ap\text{-}concat} : \prod_{l,m,n:A} \prod_{p:l=m} \prod_{q:m=n} \mathtt{ap}_f(p \cdot q) = \mathtt{ap}_f(p) \cdot \mathtt{ap}_f(q)$$

$$\Gamma \vdash \underline{\quad\quad} : \prod_{m,n:A} \prod_{p:m=n} \mathtt{ap}_{\mathtt{id}_A}(p) = p$$

$$\Gamma \vdash \mathtt{ap\text{-}comp} : \prod_{m,n:A} \prod_{p:m=n} \mathtt{ap}_{g \circ f}(p) = \mathtt{ap}_g(\mathtt{ap}_f(p))$$

Task 1. *Implement* ap-concat *and* ap-comp.

$$\Gamma \vdash \mathtt{ap\text{-}concat} : \prod_{l,m,n:A} \prod_{p:l=m} \prod_{q:m=n} \mathtt{ap}_f(p \cdot q) = \mathtt{ap}_f(p) \cdot \mathtt{ap}_f(q)$$

$$\Gamma \vdash \mathtt{ap\text{-}comp} : \prod_{m,n:A} \prod_{p:m=n} \mathtt{ap}_{g \circ f}(p) = \mathtt{ap}_g(\mathtt{ap}_f(p))$$

You may assume that $\mathtt{refl}(m) \cdot p \equiv p$, $\mathtt{refl}(m)^{-1} \equiv \mathtt{refl}(m)$, *and* $\mathtt{ap}_f(\mathtt{refl}(m)) \equiv \mathtt{refl}(f(m))$. **(Hint)** *Make sure that your motives are well-typed as you did for the associativity in Homework 3.*

Solution: All the functions:

$$\begin{aligned}
&\lambda m.\mathtt{refl}_{f(m)=f(m)}(\mathtt{refl}_B(f(m)))\\
&\lambda mnp.\mathsf{J}[m.n.p.\mathtt{ap}_f(p^{-1}) = \mathtt{ap}_f(p)^{-1}](p, x.\mathtt{refl}_{f\,m=f\,m}(\mathtt{refl}_B(f(m))))\\
\mathtt{ap\text{-}concat} :\equiv\ &\lambda lmnp.\mathsf{J}[l.m.p.\textstyle\prod_q \mathtt{ap}_f(p \cdot q) = \mathtt{ap}_f(p) \cdot \mathtt{ap}_f(q)](p, m.\lambda q.\mathtt{refl}_{f(m)=f(n)}(\mathtt{ap}_f(q)))\\
&\lambda mnp.\mathsf{J}[m.n.p.\mathtt{ap}_{\mathtt{id}_A}(p) = p](p, m.\mathtt{refl}_{m=m}(\mathtt{refl}_A(m)))\\
\mathtt{ap\text{-}comp} :\equiv\ &\lambda mnp.\mathsf{J}[m.n.p.\mathtt{ap}_{g\circ f}(p) = \mathtt{ap}_g(\mathtt{ap}_f(p))](p, m.\mathtt{refl}_{g(f(m))=g(f(m))}(\mathtt{refl}_C(g(f(m)))))
\end{aligned}$$

2 Paths to Infinity

It's one thing to feel that you are on the right path, but it's another to think yours is the only path.[1]

One of the common tasks in HoTT is to understand the structure of the path space $\mathtt{Id}_A(m,n)$ of a particular space A, with the goal to find a seemingly more manageable type equivalent to $\mathtt{Id}_A(m,n)$, which is to show that

$$\prod_{x,y:A} \mathtt{Id}_A(x,y) \simeq F(x,y)$$

for some "explicit" F. Recall that the data format of an equivalence is $(f,(g,\alpha),(h,\beta))$ where α is a witness of g being a right-inverse of f and β is of h being a left-inverse.

2.1 Tops and Bottoms are Dual in Many Ways

Task 2. *Show that* $\prod_{x,y:\top} \mathtt{Id}_\top(x,y) \simeq \top$ *(where* $F(x,y) :\equiv \top$*). You do not have to justify your code.* **(Hint)** *What is the* η *rule of* $\top$*?*

Solution: $\lambda xy.(f,(g,\alpha),(h,\beta))$ with the following data:

$$\begin{aligned}
f &:\equiv \lambda_.\langle\rangle\\
g &:\equiv \lambda_.\mathtt{refl}_\top(\langle\rangle)\\
\alpha &:\equiv \lambda_.\mathtt{refl}_\top(\langle\rangle)\\
h &:\equiv \lambda_.\mathtt{refl}_\top(\langle\rangle)\\
\beta &:\equiv \lambda p.\mathsf{J}[_._.p.\mathtt{refl}_\top(\langle\rangle) = p](p, _.\mathtt{refl}_{\langle\rangle=\langle\rangle}(\mathtt{refl}_\top(\langle\rangle)))
\end{aligned}$$

[1]This quote has been contributed to Paulo Coelho on the Internet, but I could not locate the source.

Task 3. *Show that* $\prod_{x,y:\bot} \mathtt{Id}_\bot(x,y) \simeq \bot$ *(where* $F(x,y) :\equiv \bot$*). You do not have to justify your code.* **(Hint)** *Favonia can answer this with less than 30 characters.*

Solution: $\lambda xy.\mathtt{abort}[\mathtt{Id}_\bot(x,y) \simeq \bot](x)$

2.2 Pathbreaking Techniques

Now let's focus on the product type. In class we almost finished the characterization except the definition of $\mathtt{ap}^2$ and some critical lemmas. As a reminder, the goal is to show that

$$\prod_{x,y:A\times B} \mathtt{Id}_{A\times B}(x,y) \simeq \mathtt{Id}_A(\mathtt{fst}(x),\mathtt{fst}(y)) \times \mathtt{Id}_B(\mathtt{snd}(x),\mathtt{snd}(y)).$$

There are many, many ways to define $\mathtt{ap}^2$ but we will only cover two special ones. Their particularity is that the Eckmann-Hilton theorem can be trivially derived from the equivalence between them! One of them is shown below, assuming that $f : C \to D \to E$, $p : c_1 =_C c_2$, and $q : d_1 =_D d_2$ in the current context.

$$\mathtt{ap}^2_f(p,q) :\equiv \mathtt{ap}_{\lambda c.f(c)(d_1)}(p) \cdot \mathtt{ap}_{f(c_2)}(q)$$

Task 4. *Find another one which is "symmetric" to the above implementation, where* q *comes before* p *instead of* p *coming before* q*. (We will refer to this implementation as* $\mathtt{ap}'^2_f$*.)*

Solution: $\mathtt{ap}'^2_f(p,q) :\equiv \mathtt{ap}_{f(c_1)}(q) \cdot \mathtt{ap}_{\lambda c.f(c)(d_2)}(p)$

In order to complete the characterization, for every $x, y : A \times B$ we need to prepare the data $(f,(g,\alpha),(h,\beta))$ for the equivalence. We already implemented f, g and h as follows:

$$\begin{array}{lcl} f & :\equiv & \lambda p.\langle \mathtt{ap}_{\mathtt{fst}}(p), \mathtt{ap}_{\mathtt{snd}}(p)\rangle \\ g & :\equiv & \lambda q.\mathtt{ap}^2_{\lambda ab.\langle a,b\rangle}(\mathtt{fst}(q),\mathtt{snd}(q)) \\ h & :\equiv & g \end{array}$$

The remaining are α and β. The β of type

$$\prod_{p:\mathtt{Id}_{A\times B}(x,y)} h(f(p)) = p$$

is relatively easy because the domain is $\mathtt{Id}_{A\times B}(x,y)$ and J is directly applicable. Let's look at the more interesting α of type

$$\prod_{q:\mathtt{Id}_A(\mathtt{fst}(x),\mathtt{fst}(y))\times\mathtt{Id}_B(\mathtt{snd}(x),\mathtt{snd}(y))} f(g(q)) = q.$$

Task 5. *Implement α. You may assume that $\mathtt{refl}(m) \cdot p \equiv p$ and $\mathtt{ap}_f(\mathtt{refl}(m)) \equiv \mathtt{refl}(f(m))$, as usual.* **(Hint)** *Generalize the theorem so that you have free endpoints instead of $\mathtt{fst}(x)$ or $\mathtt{snd}(y)$.*

Solution:

$\alpha :\equiv \lambda q.\mathsf{J}[x_1.y_1.q_1.f(\mathtt{ap}^2_{\lambda ab.\langle a,b\rangle}(q_1, \mathtt{ap}_{\mathtt{snd}}(q))) = \langle q_1, \mathtt{ap}_{\mathtt{snd}}(q)\rangle](\mathtt{fst}(q), z_1.$
$J[x_2.y_2.q_2.f(\mathtt{ap}_{\lambda b.\langle z_1,b\rangle}(q_2)) = \langle \mathtt{refl}(z_1), q_2\rangle](\mathtt{snd}(q), z_2.\mathtt{refl}(\langle \mathtt{refl}(z_1), \mathtt{refl}(z_2)\rangle)))$

2.3 The Mysterious Case of Eckmann and Hilton

As mentioned above, the equivalence between $\mathtt{ap}^2$ and $\mathtt{ap}'^2$ you gave in Task 4 directly leads to the Eckmann-Hilton theorem,

$$\prod_{(m:A)} \prod_{(p,q:\mathtt{refl}_A(m)=\mathtt{refl}_A(m))} p \cdot q = q \cdot p,$$

which asserts the abelianness of higher groups induced by loops at $\mathtt{refl}_A(m)$ and path concatenation. We will prove this in this subsection.

Note that this is the first "big" theorem in homework assignments and the code could easily become unreadable if Favonia did not divide the theorem into multiple tasks. The following optional text shows one way to write readable code even when the proof is ridiculously long. (You do not have to follow the style.)

The code of type $a = b$ often involves a long chain of transitivity from a to b through lots of intermediate points. If we just write down paths $p_0 \cdot (p_1 \cdot \cdots p_{n-1} \cdots)$ as the code, no one can understand the logic. A better solution is to write down *both* the proof *and* the points in a clear manner. (In proof-irrelevant mathematics, one simply write down all the points as $a_0 = a_1 = \cdots = a_n$.) Here we will introduce to you the style adopted in the current `HoTT-Agda` library (containing computer-checked HoTT proofs written for the proof assistant `Agda`). The general form of this style is

$$
\begin{array}{l}
a_0 \\
\quad =\!\langle\, p_0 \,\rangle \\
a_1 \\
\quad \vdots \\
a_{n-1} \\
\quad =\!\langle\, p_{n-1} \,\rangle \\
a_n \\
\quad \square
\end{array}
$$

where p_i is a path showing that a_i and a_{i+1} are equal. For example, the following code snippet demonstrates the associativity among four paths, assuming that the lemma $\texttt{concat-assoc}(p)(q)(r)$ inhabits the type $(p \cdot q) \cdot r = p \cdot (q \cdot r)$.

$$
\begin{array}{l}
((p \cdot q) \cdot r) \cdot s \\
\quad =\!\langle\, \texttt{ap}_{\lambda x. x \cdot s}(\texttt{concat-assoc}(p)(q)(r)) \,\rangle \\
(p \cdot (q \cdot r)) \cdot s \\
\quad =\!\langle\, \texttt{concat-assoc}(p)(q \cdot r)(s) \,\rangle \\
p \cdot ((q \cdot r) \cdot s) \\
\quad =\!\langle\, \texttt{ap}_{\lambda x. p \cdot x}(\texttt{concat-assoc}(q)(r)(s)) \,\rangle \\
p \cdot (q \cdot (r \cdot s)) \\
\quad \square
\end{array}
$$

One can check that every path p_i in the above example indeed has the type $a_i = a_{i+1}$. If a_i and a_{i+1} are definitionally equal but you nonetheless want to write them out for clarity, a trivial path $\texttt{refl}(a_i)$ can be used. This style essentially marks the endpoints of every path in the chain of transitivity, but is clearer than using the primitive `trans` directly (as you did in Homework 3). As a side note, the general form can be implemented as $p_0 \cdot (p_1 \cdot \cdots (p_{n-1} \cdot \mathtt{refl}(a_n)) \cdots)$, and different implementations can give rise to different computational behaviors.

Task 6. *Assuming* $p, q : \mathtt{refl}(m) = \mathtt{refl}(m)$, *show that*

$$\mathtt{ap}^2_{\lambda xy.x \cdot y}(p, q) = p \cdot q$$

with access to the following lemmas.

$$\mathtt{ap\text{-}id}(p) : \mathtt{ap}_{\mathtt{id}}(p) = p$$
$$\mathtt{lemma}_1(p) : \mathtt{ap}_{\lambda x.x \cdot \mathtt{refl}(n)}(p) = p$$

(Hint) $\lambda x.\mathtt{refl}(n) \cdot x \equiv \mathtt{id}$.

CLARIFICATION (2013/11/11 11AM): The $\mathtt{lemma}_1$ in Task 6 is taking a path of type $\mathtt{refl}(n) = \mathtt{refl}(n)$ for some n, not a path with generic endpoints, as this is what you need in proving the theorem. It is a good exercise to find out a generalized theorem which can be used directly as the motive of J.

Solution: $\mathtt{code}_1(p)(q) :\equiv \mathtt{ap}^2_{\lambda xy.x \cdot y}(\mathtt{lemma}_1(p), \mathtt{ap\text{-}id}(q))$

Bonus Task 1. *Similarly, show that* $\mathtt{ap}'^2_{\lambda xy.x \cdot y}(p, q) = q \cdot p$.

Solution: $\mathtt{code}_2(p)(q) :\equiv \mathtt{ap}^2_{\lambda xy.x \cdot y}(\mathtt{ap\text{-}id}(q), \mathtt{lemma}_1(q))$

Task 7. *Prove the Eckmann-Hilton theorem with the following lemma:*

$$\mathtt{lemma}_2(f)(p)(q) : \mathtt{ap}^2_f(p, q) = \mathtt{ap}'^2_f(p, q).$$

Solution:

$$\lambda mpq.\mathtt{code}_1(p)(q)^{-1} \cdot (\mathtt{lemma}_2(\lambda pq.p \cdot q)(p)(q) \cdot \mathtt{code}_2(p)(q))$$

3 Equivalence is an Equivalence

Please finish the following tasks without using the Univalence Axiom.

Task 8. ***Reflexivity.*** *Show that $A \simeq A$ for any $A : \mathcal{U}$.*

Solution: $(f, (g, \alpha), (h, \beta))$ with the following components:

$$\begin{aligned}
f &:\equiv \mathtt{id}_A \\
g &:\equiv \mathtt{id}_A \\
\alpha &:\equiv \lambda a.\mathtt{refl}_A(a) \\
h &:\equiv \mathtt{id}_A \\
\beta &:\equiv \lambda a.\mathtt{refl}_A(a)
\end{aligned}$$

Task 9. ***Symmetry.*** *Assuming you have the data $(f, (g, \alpha), (h, \beta))$ of type $A \simeq B$, show that $B \simeq A$. You may assume that $\mathsf{qinv}(f)$ with the data (g', α', β') is also available, but it is fun to finish this task without using it.*

Solution: There are many ways without using $\mathsf{qinv}(f)$. This is not an exhaustive list.

- Picking g.

$$\begin{aligned}
f' &:\equiv g \\
g' &:\equiv f \\
\alpha' &:\equiv \lambda a.(\beta(g(f(a)))^{-1}) \cdot (\mathtt{ap}_h(\alpha(f(a)))) \cdot \beta(a)) \\
h' &:\equiv f \\
\beta' &:\equiv \alpha
\end{aligned}$$

- Picking h.

$$\begin{aligned}
f' &:\equiv h \\
g' &:\equiv f \\
\alpha' &:\equiv \beta \\
h' &:\equiv f \\
\beta' &:\equiv \lambda b.(\alpha(f(g(b)))^{-1}) \cdot (\mathtt{ap}_f(\beta(g(b))) \cdot \alpha(b))
\end{aligned}$$

- Balanced choice.

$$
\begin{aligned}
f' &:\equiv h \circ (f \circ g) \\
g' &:\equiv f \\
\alpha' &:\equiv \lambda a.\mathtt{ap}_h(\alpha(f(a))) \cdot \beta(a) \\
h' &:\equiv f \\
\beta' &:\equiv \lambda b.\mathtt{ap}_f(\beta(g(b))) \cdot \alpha(b)
\end{aligned}
$$

Here's one way to use $\mathsf{qinv}(f)$.

$$
\begin{aligned}
f'' &:\equiv g' \\
g'' &:\equiv f \\
\alpha'' &:\equiv \beta' \\
h'' &:\equiv f \\
\beta'' &:\equiv \alpha'
\end{aligned}
$$

Task 10. ***Transitivity.*** *Assuming you have the data $(f_1, (g_1, \alpha_1), (h_1, \beta_1))$ of type $A \simeq B$ and $(f_2, (g_2, \alpha_2), (h_2, \beta_2))$ of type $B \simeq C$ at hand, show that $A \simeq C$.*

Solution:

$$
\begin{aligned}
f &:\equiv f_2 \circ f_1 \\
g &:\equiv g_1 \circ g_2 \\
\alpha &:\equiv \lambda b.\mathtt{ap}_{f_2}(\alpha_1(g_2(b))) \cdot \alpha_2(b) \\
h &:\equiv h_1 \circ h_2 \\
\beta &:\equiv \lambda a.\mathtt{ap}_{h_1}(\beta_2(f_1(a))) \cdot \beta_1(a)
\end{aligned}
$$