2-ADJOINT EQUIVALENCES IN HOMOTOPY TYPE THEORY

DANIEL CARRANZA, JONATHAN CHANG, KRZYSZTOF KAPULKIN, AND RYAN SANDFORD

Department of Mathematics, University of Western Ontario
URL: `http://daniel-carranza.github.io/`
e-mail address: dcarran@uwo.ca

Department of Combinatorics and Optimization, University of Waterloo
e-mail address: jchan767@uwo.ca

Department of Mathematics, University of Western Ontario
e-mail address: kkapulki@uwo.ca

Department of Computer Science, University of Western Ontario
e-mail address: rsandfo@uwo.ca

ABSTRACT. We introduce the notion of (half) 2-adjoint equivalences in Homotopy Type Theory and prove their expected properties. We formalized these results in the Lean Theorem Prover.

INTRODUCTION

There are numerous notions of equivalence in homotopy type theory: bi-invertible maps, contractible maps, and half adjoint equivalences. Other natural choices, such as quasi-invertible maps and adjoint equivalences, while *logically* equivalent to the above, are not propositions, making them unsuitable to serve as *the* definition of an equivalence. One can use a simple semantical argument, which in essence comes down to analyzing subcomplexes of the nerve of the groupoid $(0 \cong 1)$, to see why some definitions work and others do not. The conclusion here is that while the definition as a "half n-adjoint equivalence" gives us a proposition, the definition as a "(full) n-adjoint equivalence" does not.

In this paper, we take the first step towards expressing these results internally in type theory, putting special emphasis on their formalization. In particular, we revisit the notions of a quasi-invertible map, a half adjoint equivalence, and an adjoint equivalence, giving the formal proofs of their expected properties. Our proofs are more modular than those given in [Uni13], and help improve efficiency. We then turn our attention to corresponding notions arising from 2-adjunctions, namely half 2-adjoint equivalences and (full) 2-adjoint equivalences, and show that while the former is always a proposition, the latter fails to be one in general.

As indicated above, the results proven here certainly will not come as a surprise to experts and they constitute merely the first step towards understanding general n-adjoint

equivalences. One can therefore envision future work in which the notions of 3-, 4-, … and, more generally, n-adjoint equivalence are studied. We have chosen not to pursue this direction, simply because the corresponding notions of 3-, 4-, and n-adjunction have not — to our knowledge — received rigorous treatment in literature on category theory. In particular, it is not immediately clear what the higher-dimensional analog of the coherences appearing in the swallow-tail identities ought to be (cf. Coh η in Definition 3.2). Having said that, we believe that the approach developed here can serve as a blueprint for proving analogous properties of n-adjoint equivalences when these notions are introduced.

These results have been formalized using the Lean Theorem Prover, version 3.4.2 (`https://github.com/leanprover/lean`) as part of the HoTT in Lean 3 library (`https://github.com/gebner/hott3`); the formalization consists of 528 lines of code across 3 files and may be found in the directory `hott3/src/hott/types/2_adj`. We write `file/name` for a newly-formalized result, where `file` denotes the file it is found in and `name` denotes the name of the formal proof in the code.

ORGANIZATION.

Section 1 recalls the necessary background on equivalences which will be used throughout. Section 2 introduces new formal proofs that the types of quasi-inverses and adjoint equivalences are not propositions. Note that specific examples where this fails are presented, but not formally proven since the current version of the HoTT in Lean 3 library does not contain induction principles for the higher inductive types S^1 and S^2. Section 3 introduces half 2-adjoint equivalences, which are propositions containing the data of adjoint equivalences, as well as 2-adjoint equivalences, which are non-propositions related to both quasi-inverses and adjoint equivalences.

1. PRELIMINARIES

We largely adopt the notation of [Uni13], with additional and differing notation stated here. We notate the `ap` function for $f : A \to B$ by

$$f[-] : (x = y) \to (fx = fy).$$

For ap_2, the action of f on 2-dimensional paths, we write

$$f[\![-]\!] : (p = q) \to (f[p] = f[q]).$$

We write refl for the homotopy $\lambda x.\mathsf{refl}_x : \mathsf{id}_A \sim \mathsf{id}_A$. For a homotopy $H : f \sim g$ between dependent functions $f, g : \prod_{x:A} Bx$ and a non-dependent function $h : C \to A$, we write

$$H_h : fh \sim gh$$

for the composition of H and h. If $f, g : A \to B$ are non-dependent and we instead have $h : B \to C$, we write

$$h[H] : hf \sim hg$$

for the composition of $h[-]$ and H. Given an additional homotopy $H' : f \sim g$ and $\alpha : H \sim H'$, we similarly write

$$h[\![\alpha]\!] : h[H] \sim h[H']$$

for the composition of $h[\![-]\!]$ and α. Lastly, for $H : f \sim g$ and $H' : g \sim h$, we write transitivity of homotopies as

$$H \cdot H' : f \sim h$$

in path-concatenation order.

Definition 1.1 (`adj/qinv`, `adj/is_hadj_l`). A function $f : A \to B$

(1) has a *quasi-inverse* if the following type is inhabited:

$$\mathsf{qinv}\, f :\equiv \sum_{g:B\to A} gf \sim \mathsf{id}_A \times fg \sim \mathsf{id}_B.$$

(2) is a *half-adjoint equivalence* if the following type is inhabited:

$$\mathsf{ishadj}\, f :\equiv \sum_{g:B\to A} \; \sum_{\eta:gf\sim\mathsf{id}_A} \; \sum_{\varepsilon:fg\sim\mathsf{id}_B} f[\eta] \sim \varepsilon_f.$$

(3) is a *left half-adjoint equivalence* if the following type is inhabited:

$$\mathsf{ishadjl}\, f :\equiv \sum_{g:B\to A} \; \sum_{\eta:gf\sim\mathsf{id}_A} \; \sum_{\varepsilon:fg\sim\mathsf{id}_B} \eta_g \sim g[\varepsilon].$$

For types $A, B : \mathcal{U}$, the type of equivalences between A and B is:

$$A \simeq B :\equiv \sum_{f:A\to B} \mathsf{ishadj}\, f.$$

Theorem 1.2 [Uni13, Lem. 4.2.2, Thms. 4.2.3, 4.2.13]. *For $f : A \to B$, there are maps*

$$\mathsf{ishadj}\, f \xleftrightarrow{\;\simeq\;} \mathsf{ishadjl}\, f$$

$$\mathsf{qinv}\, f$$

where the top two types are propositions. □

The perhaps most intuitive definition of an equivalence between types $A, B : \mathcal{U}$ is that of a quasi-inverse. However, as this type is not a proposition, we define equivalences to be half adjoint equivalences. Since both half and left half adjoint equivalences are propositional types, one could also define the type of equivalences to be left half adjoint equivalences.

With a well-behaved notion of equivalence, we present the remaining lemmas to be used throughout.

Lemma 1.3 (Equivalence Induction/Univalence, [Uni13, Cor. 5.8.5]). *Given $D : \prod_{A,B:\mathcal{U}}(A \simeq B) \to \mathcal{U}$ and $d : \prod_{A:\mathcal{U}} D(A, A, \mathsf{id}_A)$, there exists*

$$f : \prod_{A,B:\mathcal{U}} \; \prod_{e:A\simeq B} D(A, B, e)$$

such that $f(A, A, \mathsf{id}_A) = d(A)$ for all $A : \mathcal{U}$. □

Lemma 1.4 (`prelim/sigma_hty_is_contr`, [Uni13, Cor. 5.8.6, Thm. 5.8.4]). *Given $f : A \to B$, the types*

$$\sum_{g:B\to A} f \sim g \;\text{ and }\; \sum_{g:B\to A} g \sim f$$

are both contractible with center (f, refl_f). □

Lemma 1.5 [Uni13, Lem. 4.2.5]**.** *For any* $f : A \to B$, $y : B$ *and* $(x, p), (x', p') : \mathsf{fib}_f \, y$, *we have*

$$(x, p) = (x', p') \simeq \sum_{\gamma : x = x'} p = f[\gamma] \cdot p'. \qquad \square$$

Lemma 1.6 [Uni13, Thm. 4.2.6]**.** *If* $f : A \to B$ *is a half-adjoint equivalence, then for any* $y : B$ *the fiber* $\mathsf{fib}_f \, y$ *is contractible.* $\qquad \square$

2. Quasi-inverses and adjoint equivalences, revisited

We present a proof that the type of quasi-inverses is not a proposition, using Lemma 1.4 for increased modularity over the proof presented in [Uni13, Lem. 4.1.1].

Theorem 2.1 (`adj/qinv_equiv_pi_eq`)**.** *Given* $f : A \to B$ *such that* $\mathsf{ishadj} \, f$ *is inhabited, we have*

$$\mathsf{qinv} \, f \simeq \prod_{x : A} x = x.$$

Proof. By Equivalence Induction 1.3, it suffices to show $\mathsf{qinv} \, \mathsf{id}_A \simeq \prod_{x:A} x = x$. Observe that

$$\mathsf{qinv} \, \mathsf{id}_A \equiv \sum_{g : A \to A} g \sim \mathsf{id}_A \times g \sim \mathsf{id}_A$$

$$\simeq \sum_{g : A \to A} \sum_{\eta : g \sim \mathsf{id}_A} g \sim \mathsf{id}_A$$

$$\simeq \sum_{u : \sum\limits_{g : A \to A} g \sim \mathsf{id}_A} \mathsf{pr}_1 \, u \sim \mathsf{id}_A$$

$$\simeq \mathsf{id}_A \sim \mathsf{id}_A \qquad (2.1)$$

$$\equiv \prod_{x : A} x = x,$$

where (2.1) follows from Lemma 1.4 (the type $\sum\limits_{g : A \to A} g \sim \mathsf{id}_A$ is contractible with center $(\mathsf{id}_A, \mathsf{refl})$). $\qquad \square$

This result implies that any type with non-trivial π_1 may be used to construct non-trivial inhabitants of this type. For instance, since $\pi_1(S^1) = \mathbb{Z}$, we have:

Corollary 2.2. *The type* $\mathsf{qinv} \, \mathsf{id}_{S^1}$ *is not a proposition.* $\qquad \square$

Conceptually, this proof takes the pair (g, η) and uses Lemma 1.4 to contract it so that only one homotopy remains. This differs from the proof in [Uni13], which uses function extensionality to write the homotopies as paths and contracts using based path induction. This proof modularizes the proof in [Uni13] by packaging function extensionality and rewriting of contractible types into one result, simplifying both the proof and the formalization.

Thus, the type of half and left half adjoint equivalences each append an additional coherence to contract with the remaining homotopy. However, appending both coherences gives us a non-proposition.

Definition 2.3 (`adj/adj`). Given $f : A \to B$, the structure of an *adjoint equivalence* on f is the type:

$$\mathsf{adj}\, f :\equiv \sum_{g:B\to A} \sum_{\eta:gf\sim\mathsf{id}_A} \sum_{\varepsilon:fg\sim\mathsf{id}_B} f[\eta] \sim \varepsilon_f \times \eta_g \sim g[\varepsilon].$$

Theorem 2.4 (`adj/adj_equiv_pi_refl_eq`). *Given $f : A \to B$ such that $\mathsf{ishadj}\, f$ is inhabited, we have*

$$\mathsf{adj}\, f \simeq \prod_{x:A}(\mathsf{refl}_x = \mathsf{refl}_x).$$

Proof. By Equivalence Induction 1.3, it suffices to show $\mathsf{adj}\,\mathsf{id}_A \simeq \prod_{x:A} \mathsf{refl}_x = \mathsf{refl}_x$. Observe that

$$
\begin{aligned}
\mathsf{adj}\,\mathsf{id}_A &\equiv \sum_{g:A\to A} \sum_{\eta:g\sim\mathsf{id}_A} \sum_{\varepsilon:g\sim\mathsf{id}_A} \mathsf{id}_A[\eta] \sim \varepsilon \times \eta_g \sim g[\varepsilon] \\
&\simeq \sum_{\varepsilon:\mathsf{id}_A\sim\mathsf{id}_A} \mathsf{refl} \sim \varepsilon \times \mathsf{refl} \sim \mathsf{id}_A[\varepsilon] & (2.2) \\
&\simeq \sum_{\varepsilon:\mathsf{id}_A\sim\mathsf{id}_A} \sum_{\tau:\mathsf{refl}\sim\varepsilon} \mathsf{refl} \sim \mathsf{id}_A[\varepsilon] \\
&\simeq \sum_{u:\sum_{\varepsilon:\mathsf{id}_A\sim\mathsf{id}_A}\mathsf{refl}\sim\varepsilon} \mathsf{refl} \sim \mathsf{id}_A[\mathsf{pr}_1\, u] \\
&\simeq \mathsf{refl} \sim \mathsf{id}_A[\mathsf{refl}] & (2.3) \\
&\equiv \prod_{x:A}(\mathsf{refl}_x = \mathsf{refl}_x).
\end{aligned}
$$

The equivalence (2.2) comes from the equivalence in Theorem 2.1, where the pair (g, η) contracts to $(\mathsf{id}_A, \mathsf{refl})$. The equivalence (2.3) follows from Lemma 1.4. □

This result implies that any type with non-trivial π_2 may be used to construct non-trivial inhabitants of this type. In particular, $\pi_2(S^2) = \mathbb{Z}$ proves the following:

Corollary 2.5. *The type $\mathsf{adj}\,\mathsf{id}_{S^2}$ is not a proposition.* □

This is a solution to Exercise 4.1 in [Uni13]. As before, this proof uses Lemma 1.4 to contract the pairs (g, η) and (ε, τ) so that a single homotopy remains. Trying to apply path induction directly requires an equivalence which writes each homotopy as an equality; a formal proof using function extensionality for such an equivalence along with path induction reaches 60 lines of code (varying by format, syntax choice, etc.). By modularizing the case of qinv, this proof is reduced to manipulating Σ-types and applying Lemma 1.4 twice, with the formal proof in the library being 23 lines of code.

3. 2-ADJOINT EQUIVALENCES

As in the case of qinv, we expect there is an additional coherence that may be appended to the type $\mathsf{adj}\, f$ to create a proposition. To define this coherence, we use the following homotopy:

Lemma 3.1 (`two_adj/nat_coh`). *Given $f : A \to B$ and $g : B \to A$ with a homotopy $H : gf \sim \mathrm{id}_A$, we have a homotopy*

$$\mathsf{Coh}\, H : H_{gf} \sim g[f[H]]$$

such that

$$\mathsf{Coh}\, \mathsf{refl} \equiv \mathsf{refl}_{\mathsf{refl}} : \mathsf{refl} \sim \mathsf{refl}.$$

Proof. Fix $x : A$. We have

$$
\begin{aligned}
H_{g(fx)} &= (gf)[H_x] \\
&= g[f[H_x]],
\end{aligned}
$$

where the first equality holds by naturality and the second holds by functoriality of $g[-]$. □

With this, we define the type of half 2-adjoint equivalences.

Definition 3.2 (`two_adj/is_two_hae`). *A function $f : A \to B$ is a half 2-adjoint equivalence if the following type is inhabited:*

$$\mathsf{ish2adj}\, f :\equiv \sum_{g:B\to A} \sum_{\eta:gf\sim\mathrm{id}_A} \sum_{\varepsilon:fg\sim\mathrm{id}_B} \sum_{\tau:f[\eta]\sim\varepsilon_f} \sum_{\theta:\eta_g\sim g[\varepsilon]} \mathsf{Coh}\, \eta \cdot g[\![\tau]\!] \sim \theta_f.$$

In parallel with adjoint equivalences, we give a definition which uses an alternate coherence.

Definition 3.3 (`two_adj/is_two_hae_l`). *A function $f : A \to B$ is a left half 2-adjoint equivalence if the following type is inhabited:*

$$\mathsf{ish2adjl}\, f :\equiv \sum_{g:B\to A} \sum_{\eta:gf\sim\mathrm{id}_A} \sum_{\varepsilon:fg\sim\mathrm{id}_B} \sum_{\tau:f[\eta]\sim\varepsilon_f} \sum_{\theta:\eta_g\sim g[\varepsilon]} \tau_g \cdot \mathsf{Coh}\, \varepsilon \sim f[\![\theta]\!].$$

To show the type of half 2-adjoint equivalences is a proposition, we prove the following lemma:

Lemma 3.4 (`two_adj/r2coh_equiv_fib_eq`). *Given $f : A \to B$ with $(g, \eta, \varepsilon, \theta) : \mathsf{ishadjl}\, f$, we have*

$$\sum_{\tau:f[\eta]\sim\varepsilon_f} \mathsf{Coh}\, \eta \cdot g[\![\tau]\!] \sim \theta_f \simeq \prod_{x:A} (f[\eta_x], \mathsf{Coh}\, \eta_x \cdot \theta_{fx}) = \left(\varepsilon_{fx}, \mathsf{refl}_{g[\varepsilon_{fx}]}\right),$$

where $(f[\eta_x], \mathsf{Coh}\, \eta_x \cdot \theta_{fx}), \left(\varepsilon_{fx}, \mathsf{refl}_{g[\varepsilon_{fx}]}\right) : \mathsf{fib}_{g[-]}\, g[\varepsilon_{fx}].$

Proof. We have

$$\sum_{\tau:f[\eta]\sim\varepsilon_f} \mathsf{Coh}\, \eta \cdot g[\![\tau]\!] \sim \theta_f \equiv \sum_{\tau:\prod_{x:A} f[\eta_x]=\varepsilon_{fx}} \prod_{x:A} \mathsf{Coh}\, \eta_x \cdot g[\![\tau_x]\!] = \theta_{fx},$$

$$\simeq \prod_{x:A} \sum_{\tau':f[\eta_x]=\varepsilon_{fx}} \mathsf{Coh}\, \eta_x \cdot g[\![\tau']\!] = \theta_{fx} \tag{3.1}$$

$$\simeq \prod_{x:A} \sum_{\tau':f[\eta_x]=\varepsilon_{f(x)}} \mathsf{Coh}\, \eta_x^{-1} \cdot \theta_{fx} = g[\![\tau']\!] \tag{3.2}$$

$$\simeq \prod_{x:A} \left(f[\eta_x], (N_\eta)^{-1} \cdot \theta_{f(x)}\right) = \left(\varepsilon_{f(x)}, \mathsf{refl}_{g[\varepsilon_{f(x)}]}\right). \tag{3.3}$$

The equivalence (3.1) holds by the Type-Theoretic Axiom of Choice, (3.2) is a rearrangment of equality, and (3.3) holds by Lemma 1.5. □

Lemma 3.5 (`two_adj/is_contr_r2coh`). *Given $f : A \to B$ with $(g, \eta, \varepsilon, \theta)$: ishadj f, the type*

$$\sum_{\tau : f[\eta] \sim \varepsilon_f} \mathsf{Coh}\, \eta \cdot g[\![\tau]\!] \sim \theta_f$$

is contractible.

Proof. By Lemma 3.4 and contractibility of Π-types, it suffices to fix $x : A$ and show the type

$$\left(f[\eta_x], \mathsf{Coh}\, \eta_x^{-1} \cdot \theta_{fx}\right) = \left(\varepsilon_{fx}, \mathsf{refl}_{g[\varepsilon_{fx}]}\right)$$

is contractible. Since g is an equivalence, $g[-]$ is also an equivalence. By Lemma 1.6, the type $\mathsf{fib}_{g[-]}(g[\varepsilon_{fx}])$ is contractible, so its equality type is also contractible. $\qquad\square$

Theorem 3.6 (`two_adj/is_prop_is_two_hae`). *For any $f : A \to B$, the type ish2adj f is a proposition.*

Proof. It suffices to assume e : ish2adj f and show this type is contractible. Observe that

$$\begin{aligned}
\mathsf{ish2adj}\, f &\equiv \sum_{g : B \to A} \sum_{\eta : gf \sim \mathsf{id}_A} \sum_{\varepsilon : fg \sim \mathsf{id}_B} \sum_{\tau : f[\eta] \sim \varepsilon_f} \sum_{\theta : \eta_g \sim g[\varepsilon]} \mathsf{Coh}\, \eta \cdot g[\![\tau]\!] \sim \theta_f \\
&\simeq \sum_{g : B \to A} \sum_{\eta : gf \sim \mathsf{id}_A} \sum_{\varepsilon : fg \sim \mathsf{id}_B} \sum_{\theta : \eta_g \sim g[\varepsilon]} \sum_{\tau : f[\eta] \sim \varepsilon_f} \mathsf{Coh}\, \eta \cdot g[\![\tau]\!] \sim \theta_f \\
&\simeq \sum_{(g, \eta, \varepsilon, \theta) : \mathsf{ishadjl}\, f} \sum_{\tau : f[\eta] \sim \varepsilon_f} \mathsf{Coh}\, \eta \cdot g[\![\tau]\!] \sim \theta_f \\
&\simeq \sum_{\tau : f[\eta_0] \sim (\varepsilon_0)_f} \mathsf{Coh}\, \eta_0 \cdot g_0[\![\tau]\!] \sim (\theta_0)_f.
\end{aligned}$$

The last equivalence holds since ishadjl$f$ is contractible (it is a proposition and inhabited by e after discarding coherences); we write $(g_0, \eta_0, \varepsilon_0, \theta_0)$: ishadjl$f$ for its center of contraction. This final type is contractible by Lemma 3.5, therefore ish2adj$f$ is contractible. $\qquad\square$

Parallels of these proofs are used to obtain similar results about left half two-adjoint equivalences as well.

Lemma 3.7 (`two_adj/is_contr_l2coh`). *Given $f : A \to B$ with $(g, \eta, \varepsilon, \tau)$: ishadj$f$, the type*

$$\sum_{\theta : \eta_g \sim g[\varepsilon]} \tau_g \cdot \mathsf{Coh}\, \varepsilon \sim f[\![\theta]\!].$$

is contractible.

Proof. Analogous to Lemma 3.5. $\qquad\square$

Theorem 3.8 (`two_adj/is_prop_is_two_hae_l`). *For $f : A \to B$, the type ish2adjl f is a proposition.*

Proof. Analogous to Theorem 3.6. $\qquad\square$

As well, either half adjoint equivalence may be promoted to the alternate half 2-adjoint equivalence.

Theorem 3.9 (`two_adj/two_adjointify`). *For $f : A \to B$, we have maps*
(1) ishadjl $f \to$ ish2adj f

(2) ishadj $f \rightarrow$ ish2adjl f

Proof. Take the missing coherences to be the centers of contraction from Lemmas 3.5 and 3.7. $\qquad \square$

This implies that an adjoint equivalence may be promoted to either half 2-adjoint equivalence.

Corollary 3.10. *For $f : A \rightarrow B$, we have maps*

(1) adj $f \rightarrow$ ish2adj f
(2) adj $f \rightarrow$ ish2adjl f

Proof. Discard either coherence and apply Theorem 3.9. $\qquad \square$

Finally, we have that the half 2-adjoint and left half 2-adjoint equivalences are logically equivalent.

Theorem 3.11 (`two_adj/two_hae_equiv_two_hae_l`). *For $f : A \rightarrow B$, we have maps*

$$\text{ish2adj } f \leftrightarrow \text{ish2adjl } f.$$

Proof. In either direction, discard coherences and apply Theorem 3.9. $\qquad \square$

We summarize the properties of these 2-adjoint equivalances with the following diagram of maps:

$$
\begin{array}{ccc}
\text{ish2adjl } f & \xleftrightarrow{\simeq} & \text{ish2adj } f \\
& \text{adj } f & \\
\text{ishadjl } f & \xleftrightarrow{\simeq} & \text{ishadj } f \\
& \text{qinv} f &
\end{array}
$$

where rows 1 and 3 are propositions.

As before, appending either one of these coherences yields a proposition, but appending both coherences yields a non-proposition once more.

Definition 3.12 (`two_adj/two_adj`). Given $f : A \rightarrow B$, the structure of a *2-adjoint equivalence* on f is the type:

$$\text{2adj } f :\equiv \sum_{g:B \rightarrow A} \sum_{\eta:gf \sim \text{id}_A} \sum_{\varepsilon:fg \sim \text{id}_B} \sum_{\tau:f[\eta] \sim \varepsilon_f} \sum_{\theta:\eta_g \sim g[\varepsilon]} \text{Coh } \eta \cdot g[\![\tau]\!] \sim \theta_f \times \tau_g \cdot \text{Coh } \varepsilon \sim f[\![\theta]\!].$$

Theorem 3.13 (`two_adj/two_adj_equiv_pi_refl_eq`). *Given $f : A \rightarrow B$ such that* ishadj f *is inhabited, we have*

$$\text{2adj } f \simeq \prod_{x:A} (\text{refl}_{\text{refl}_x} = \text{refl}_{\text{refl}_x}).$$

Proof. By Equivalence Induction 1.3, it suffices to show $\mathsf{2adj}\ \mathsf{id}_A \simeq \prod_{x:A}(\mathsf{refl}_{\mathsf{refl}_x} = \mathsf{refl}_{\mathsf{refl}_x})$. Observe that

$$\mathsf{2adj}\ \mathsf{id}_A \equiv \sum_{g:A\to A}\ \sum_{\eta:g\sim\mathsf{id}_A}\ \sum_{\varepsilon:g\sim\mathsf{id}_A}\ \sum_{\tau:\mathsf{id}_A[\eta]\sim\varepsilon}\ \sum_{\theta:\eta_g\sim g[\varepsilon]} \mathsf{Coh}\ \eta \cdot g[\![\tau]\!] \sim \theta \times \tau_g \cdot \mathsf{Coh}\ \varepsilon \sim \mathsf{id}_A[\![\theta]\!]$$

$$\simeq \sum_{\theta:\mathsf{refl}\sim\mathsf{refl}} \mathsf{Coh}\ \mathsf{refl}\cdot\mathsf{id}_A[\![\mathsf{refl}_{\mathsf{refl}}]\!] \sim \theta \times \mathsf{refl}_{\mathsf{refl}} \cdot \mathsf{Coh}\ \mathsf{refl} \sim \mathsf{id}_A[\![\theta]\!] \qquad (3.4)$$

$$\equiv \sum_{\theta:\mathsf{refl}\sim\mathsf{refl}} \mathsf{refl}_{\mathsf{refl}} \sim \theta \times \mathsf{refl}_{\mathsf{refl}} \sim \mathsf{id}_A[\![\theta]\!]$$

$$\simeq \sum_{\theta:\mathsf{refl}\sim\mathsf{refl}} \mathsf{refl}_{\mathsf{refl}} \sim \theta \times \mathsf{refl}_{\mathsf{refl}} \sim \theta$$

$$\simeq \sum_{\theta:\mathsf{refl}\sim\mathsf{refl}}\ \sum_{\mathcal{A}:\mathsf{refl}_{\mathsf{refl}}\sim\theta} \mathsf{refl}_{\mathsf{refl}} \sim \theta$$

$$\simeq \sum_{u:\ \sum_{\theta:\mathsf{refl}\sim\mathsf{refl}}\mathsf{refl}_{\mathsf{refl}}\sim\theta} \mathsf{refl}_{\mathsf{refl}} \sim \mathsf{pr}_1\ u$$

$$\simeq \mathsf{refl}_{\mathsf{refl}} \sim \mathsf{refl}_{\mathsf{refl}} \qquad (3.5)$$

$$\equiv \prod_{x:A} \mathsf{refl}_{\mathsf{refl}_x} = \mathsf{refl}_{\mathsf{refl}_x}.$$

The equivalence (3.4) is from Theorem 2.4; we contract $(g,\eta,\varepsilon,\tau)$ to $(\mathsf{id}_A,\mathsf{refl},\mathsf{refl},\mathsf{refl}_{\mathsf{refl}})$. The equivalence (3.5) is an application of Lemma 1.4. □

Once again, this result implies any type with non-trivial π_3 may be used to construct non-trivial inhabitants of this type. We know $\pi_3(S^2) = \mathbb{Z}$, which proves:

Corollary 3.14. *The type* $\mathsf{2adj}\ \mathsf{id}_{S^2}$ *is not a proposition.* □

Proving this result using function extensionality directly and path induction requires an equivalence that writes homotopies as equalities. By modularizing the case of qinv, similar to the analogous proof for adj, this result may be proven by manipulating Σ-types and applying Lemma 1.4 three times, with the formal proof in the library being 44 lines of code. As with adj, one would expect this approach to be 40 to 80 lines shorter than one which uses function extensionality directly.

References

[Uni13] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics.* http://homotopytypetheory.org/book, Institute for Advanced Study, 2013.