Higher Inductive Types as Homotopy-Initial Algebras

Kristina Sojakova

January 2014 (Revised July 2014)
CMU-CS-14-101R

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

This report is a revised version of Technical Report CMU-CS-14-101.

Support for this research was provided by the Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) through the Carnegie Mellon Portugal Program under grant NGN-44 and by the National Science Foundation Grant DMS-1001191.

Keywords: Homotopy Type Theory, higher inductive types, homotopy-initial algebras

Abstract

Homotopy Type Theory is a new field of mathematics based on the recently-discovered correspondence between Martin-Löf's constructive type theory and abstract homotopy theory. We have a powerful interplay between these disciplines - we can use geometric intuition to formulate new concepts in type theory and, conversely, use type-theoretic machinery to verify and often simplify existing mathematical proofs. Higher inductive types form a crucial part of this new system since they allow us to represent mathematical objects, such as spheres, tori, pushouts, and quotients, in the type theory. We investigate a class of higher inductive types called W-suspensions which generalize Martin-Löf's well-founded trees. We show that a propositional variant of W-suspensions, whose computational behavior is determined up to a higher path, is characterized by the universal property of being a homotopy-initial algebra. As a corollary we get that W-suspensions in the strict form are homotopy-initial.

1 Introduction

Homotopy Type Theory (HoTT) has recently generated significant interest among type theorists and mathematicians alike. It uncovers deep connections between Martin-Löf's dependent type theory ([16, 17]) and the fields of abstract homotopy theory, higher categories, and algebraic topology ([3, 5, 6, 7, 8, 11, 13, 23, 24, 25, 26]). Insights from homotopy theory are used to add new concepts to the type theory, such as the representation of various geometric objects as higher inductive types. Conversely, type theory is used to formalize and verify existing mathematical proofs using proof assistants such as Coq [21] and Agda [18]. Moreover, type-theoretic insights often help us discover novel proofs of known results which are simpler than their homotopy-theoretic versions: the calculation of $\pi_n(\mathbf{S}^n)$ ([12, 10]); the Freudenthal Suspension Theorem [22]; the Blakers-Massey Theorem [22], etc.

As a formal system, HoTT [22] is a generalization of intensional Martin-Löf Type Theory with two features motivated by abstract homotopy theory: Voevodsky's *univalence axiom* ([8, 25]) and *higher-inductive types* ([14, 19]). The slogan in HoTT is that *types are topological spaces, terms are points, and proofs of identity are paths between points*. The structure of an identity type in HoTT is thus far more complex than just consisting of reflexivity paths, despite the definition of $\mathsf{Id}_A(x, y)$ as an inductive type with a single constructor $\mathsf{refl}_A(x) : \mathsf{Id}_A(x, x)$. It is a beautiful, and perhaps surprising, fact that not only does this richer theory admit an interpretation into homotopy theory ([3], [8]) but that many fundamental concepts and results from mathematics arise naturally as constructions and theorems of HoTT.

For example, the unit circle $\mathbf{S}^1$ is defined as a higher inductive type with a point base and a loop loop based at base. It comes with a recursion principle which says that to construct a function $f : \mathbf{S}^1 \to X$, it suffices to supply a point $x : X$ and a loop based at x. The value $f(\mathsf{base})$ then computes to x. Such definitional computation rules are convenient to work with but also pose some conceptual difficulties. For instance, an alternative encoding of the circle as a higher inductive type $\mathbf{S}^1_a$ specifies two points south, north and two paths from north to south, called east and west. The recursion principle then says that in order to construct a function $f : \mathbf{S}^1_a \to X$, it suffices to supply two points $x, y : X$ and two paths between them. The values $f(\mathsf{north})$ and $f(\mathsf{south})$ then compute to x and y respectively.

We have a natural way of relating these two representations via an equivalence: in one direction, map base to north and loop to east; in the other direction, map both north and south to base and map east to loop and west to the identity path at base. Unfortunately, the types $\mathbf{S}^1$, $\mathbf{S}^1_a$ related this way, while equivalent, do *not* satisfy the same definitional laws, which poses a compatibility issue. Even more importantly, we do not have a way of *internalizing* these notions of a circle and working with them inside the type theory, since we can only talk about definitional equalities on the meta-level.

In this paper we thus study higher inductive types abstractly, as arbitrary types endowed with certain constructors and *propositional* computation behavior: in the case of $\mathbf{S}^1$, for example, we say that a type C with constructors $b : C$ and $l : c = c$ satisfies the recursion principle for a circle if for any other type X, point $x : X$ and loop based at x, there exists a function $f : C \to X$ for which there is a *path between* $f(b)$ *and* x (and which satisfies a higher coherence condition). We note that we require *no* change to the underlying type theory: the particular higher inductive type

$\mathbf{S}^1$ just becomes a specific instance of the abstract definition of a circle, one whose computation rules happen to hold definitionally.

A major advantage of types with propositional computation rules is that we can internalize the definitions and reason about them within the type theory - and in particular, use proof assistants to verify the results. In this respect, our work is complementary to [15], which gives an external, category-theoretic semantics for a certain class of higher inductives. Another advantage of propositional computation behavior is portability: relaxing the computation laws satisfied by the types $\mathbf{S}^1$ and $\mathbf{S}^1_a$ to their propositional counterparts results in two notions of a circle that are equivalent. This in particular means that any type C which is a circle in one sense is also a circle in the alternate sense. We can thus state and prove results about either of these specifications, knowing that the proofs carry over to any particular implementation - be it $\mathbf{S}^1$, $\mathbf{S}^1_a$, or a third one.

It further turns out that types with propositional rules tend to keep many of their desirable properties; for instance, it can be shown that the main result of [12], that the fundamental group of the circle is the group of integers, carries over to the case when *both* the circle and the integer types have propositional computational behavior. In addition, we can now show that higher inductive types are characterized by the universal property of being a *homotopy-initial algebra*. This notion was first introduced in [2], where an analogous result was established for the "ordinary" inductive type of well-founded trees (Martin-Löf's W-types). In the higher-dimensional setting, an *algebra* is a type X together with a number of finitary operations $f, g, h \ldots$, which are allowed to act not only on X but also on any higher identity type over X. An *algebra homomorphism* has to preserve all operations up to a higher homotopy. Finally, an algebra $\mathcal{X}$ is *homotopy-initial* if the type of homomorphisms from $\mathcal{X}$ to any other algebra $\mathcal{Y}$ is contractible.

Our main theorem is stated for a class of higher inductive types which we call W-suspensions; they generalize ordinary W-types as well as the higher inductive type $\mathbb{S}$ and others. We show that the induction principle for W-suspensions is equivalent (as a type) to homotopy-initiality. This extends the main result of [2] for "ordinary" inductive types to the important, and much more difficult, higher-dimensional case.

2 Basic Homotopy Type Theory

The core of HoTT is a dependent type theory with

- dependent pair types $\Sigma_{x:A}B(x)$ and dependent function types $\Pi_{x:A}B(x)$ (with the non-dependent versions $A \times B$ and $A \to B$). To stay consistent with the presentation in [22], we assume definitional η-conversion for functions but do not assume it for pairs.

- a cumulative hierarchy of universes $\mathcal{U}_0 : \mathcal{U}_1 : \mathcal{U}_2 : \ldots$ in the style of Russell.

- intensional identity types $\mathsf{Id}_A(x, y)$, also denoted $x =_A y$. We have the usual formation and introduction rules; the elimination and computation rules are recalled below:

$$\frac{E : \Pi_{x,y:A}\mathsf{Id}_A(x,y) \to \mathcal{U}_i \qquad d : \Pi_{x:A}E(x,x,\mathsf{refl}_A(x))}{\mathsf{J}(E,d) : \Pi_{x,y:A}\Pi_{p:\mathsf{Id}_A(x,y)}E(x,y,p)}$$

$$\frac{E : \Pi_{x,y:A}\mathsf{Id}_A(x,y) \to \mathcal{U}_i \qquad d : \Pi_{x:A}E(x,x,\mathsf{refl}_A(x)) \qquad a : A}{\mathsf{J}(E,d)(a,a,\mathsf{refl}_A(a)) \equiv d(a) : E(a,a,\mathsf{refl}_A(a))}$$

These rules are, of course, applicable in any context Γ; we follow the standard convention of omitting it. If the type $\mathsf{Id}_A(x, y)$ is inhabited, we call x and y *equal*. If we do not care about the specific equality witness, we often simply say that $x =_A y$ or if the type A is clear, $x = y$. A term $p : x =_A y$ will be often called a *path* and the process of applying the identity elimination rule will be referred to as *path induction*. Definitional equality between $x, y : A$ will be denoted as $x \equiv y : A$.

We emphasize that apart from the aforementioned identity rules, univalence, and higher inductive types there are no other rules governing the behavior of identity types - in particular, we assert neither any form of Streicher's K-rule [20] nor the identity reflection rule.

The rest of this section describes the univalence axiom and some key properties of identity types; higher inductive types are discussed in Section. 3. For a thorough exposition of homotopy type theory we refer the reader to [22].

2.1 Groupoid laws

Proofs of identity behave much like paths in topological spaces: they can be reversed, concatenated, mapped along functions, etc. Below we summarize a few of these properties:

- For any path $p : x =_A y$ there is a path $p^{-1} : y =_A x$, and we have $\mathsf{refl}_A(x)^{-1} \equiv \mathsf{refl}_A(x)$.

- For any paths $p : x =_A y$ and $q : y =_A z$ there is a path $p \cdot q : x =_A z$, and we have $\mathsf{refl}_A(x) \cdot \mathsf{refl}_A(x) \equiv \mathsf{refl}_A(x)$.

- Associativity of composition: for any paths $p : x =_A y$, $q : y =_A z$, and $r : z =_A u$ we have $(p \cdot q) \cdot r = p \cdot (q \cdot r)$.

- We have $\mathsf{refl}_A(x) \cdot p = p$ and $p \cdot \mathsf{refl}_A(y) = p$ for any $p : x =_A y$.

- For any $p : x =_A y$, $q : y =_A z$ we have $p \cdot p^{-1} = \mathsf{refl}_A(x)$, $p^{-1} \cdot p = \mathsf{refl}_A(y)$, and $(p^{-1})^{-1} = p$, $(p \cdot q)^{-1} = q^{-1} \cdot p^{-1}$.

- For any type family $P : A \to \mathcal{U}_i$ and path $p : x =_A y$ there are functions $p_*^P : P(x) \to P(y)$ and $p_P^* : P(y) \to P(x)$, called the *covariant transport* and *contravariant transport*, respectively. We furthermore have $\mathsf{refl}_A(x)_*^P \equiv \mathsf{refl}_A(x)_P^* \equiv \mathsf{id}_{P(x)}$.

- We have $(p^{-1})_*^P = p_P^*$, $(p^{-1})_P^* = p_*^P$ and $(p \cdot q)_*^P = q_*^P \circ p_*^P$, $(p \cdot q)_P^* = p_P^* \circ q_P^*$ for any family $P : A \to \mathcal{U}_i$ and paths $p : x =_A y$, $q : y =_A z$.

- For any function $f : A \to B$ and path $p : x =_A y$, there is a path $\mathsf{ap}_f(p) : f(x) =_B f(y)$ and we have $\mathsf{ap}_f(\mathsf{refl}_A(x)) \equiv \mathsf{refl}_B(f(x))$.

- We have $\mathsf{ap}_f(p^{-1}) = \mathsf{ap}_f(p)^{-1}$ and $\mathsf{ap}_f(p \cdot q) = \mathsf{ap}_f(p) \cdot \mathsf{ap}_f(q)$ for any $f : A \to B$ and $p : x =_A y$, $q : y =_A z$.

- We have $\mathsf{ap}_{g \circ f}(p) = \mathsf{ap}_g(\mathsf{ap}_f(p))$ for any $f : A \to B$, $g : B \to C$ and $p : x =_A y$.

- For a dependent function $f : \Pi_{x:A} B(x)$ and path $p : x =_A y$, there are paths $\mathsf{dap}_f(p) : p_*^B(f(x)) =_{B(y)} f(y)$ and $\mathsf{dap}^f(p) : p_B^*(f(y)) =_{B(x)} f(x)$. We also have $\mathsf{dap}_f(\mathsf{refl}_A(x)) \equiv \mathsf{dap}^f(\mathsf{refl}_A(x)) \equiv \mathsf{refl}_{B(x)}(f(x))$.

- All constructs respect propositional equality.

2.2 Homotopies between functions

A homotopy between two functions is in a sense a "natural transformation":

Definition 1. *For $f, g : \Pi_{x:A} B(x)$, we define the type*

$$f \sim g :\equiv \Pi_{a:A}(f(a) =_{B(a)} g(a))$$

and call it the type of homotopies *between f and g.*

Definition 2. *For $f : A \to B$ and $g : A' \to B$, we define the type*

$$f \sim_{\mathcal{H}} g :\equiv \Pi_{a:A}\Pi_{a':A}(f(a) =_B g(a'))$$

and call it the type of heterogeneous homotopies *between f and g.*

We now introduce some notation that will be needed later.

- For any $f : \Pi_{x:X} E(x)$, the identity homotopy on f is $\mathsf{id}_{\mathcal{H}}(f) :\equiv \lambda_{x:X} \mathsf{refl}_{E(x)}(f(x))$.

- For any $f, g : X \to Y$, $h : \Pi_{x:X} E(f(x))$, $\alpha : f \sim g$, the composition of the homotopy α and the function h is a function $\alpha \circ_{\mathcal{H}} h : \Pi_{x:X} E(g(x))$ defined as

$$\alpha \circ_{\mathcal{H}} h :\equiv \lambda_{x:X} \alpha(x)_*^E \; h(x)$$

- For any $f, g : X \to Y$, $p : x =_X y$, $\alpha : f \sim g$, there is a path

$$\mathsf{nat}(\alpha, p) : \alpha(x) \cdot \mathsf{ap}_g(p) = \mathsf{ap}_f(p) \cdot \alpha(y)$$

defined in the obvious way by induction on p and referred to as the naturality of the homotopy α. Pictorially, we have

$$\begin{array}{ccc} f(x) & \overset{\alpha(x)}{\rule{3em}{0.4pt}} & g(x) \\ \mathsf{ap}_f(p) \Big| & \mathsf{nat}(\alpha, p) & \Big| \mathsf{ap}_g(p) \\ f(y) & \underset{\alpha(y)}{\rule{3em}{0.4pt}} & g(y) \end{array}$$

- For any $f, g : \Pi_{x:X} Y(x)$, $p : x =_X y$, $\alpha : f \sim g$, there is a path

$$\mathsf{nat}_{\mathcal{F}}(\alpha, p) : \mathsf{ap}_{p_*^Y}(\alpha(x)) \cdot \mathsf{dap}_g(p) = \mathsf{dap}_f(p) \cdot \alpha(y)$$

defined in the obvious way by induction on p and referred to as the naturality of the "fibered" homotopy α. Pictorially, we have

$$\begin{array}{ccc} p_*^Y(f(x)) & \overset{\mathsf{ap}_{p_*^Y}(\alpha(x))}{\rule{4em}{0.4pt}} & p_*^Y(g(x)) \\ \mathsf{dap}_f(p) \Big| & \mathsf{nat}_{\mathcal{F}}(\alpha, p) & \Big| \mathsf{dap}_g(p) \\ f(y) & \underset{\alpha(y)}{\rule{4em}{0.4pt}} & g(y) \end{array}$$

- For any $f : X \to Z$, $g : Y \to Z$, $p : x_1 =_X x_2$, $q : y_1 =_Y y_2$, $\alpha : f \sim_{\mathcal{H}} g$, there is a path

$$\mathsf{nat}_{\mathcal{H}}(\alpha, p, q) : \alpha(x_1, y_1) \cdot \mathsf{ap}_g(q) = \mathsf{ap}_f(p) \cdot \alpha(x_2, y_2)$$

defined in the obvious way by induction on p and q and referred to as the naturality of the heterogeneous homotopy α. Pictorially, we have

$$\begin{array}{ccc} f(x_1) & \overset{\alpha(x_1, y_1)}{\rule{4em}{0.4pt}} & g(y_1) \\ \mathsf{ap}_f(p) \Big| & \mathsf{nat}_{\mathcal{H}}(\alpha, p, q) & \Big| \mathsf{ap}_g(q) \\ f(x_2) & \underset{\alpha(x_2, y_2)}{\rule{4em}{0.4pt}} & g(y_2) \end{array}$$

2.3 Truncation levels

In general, the structure of paths on a type A can be highly nontrivial - we can have many distinct *0-cells* $x, y, \ldots : A$; there can be many distinct *1-cells* $p, q, \ldots : x =_A y$; there can be many distinct *2-cells* $\gamma, \delta, \ldots : p =_{x=_A y} q$; ad infinitum. The hierarchy of truncation levels describes those types which are, informally speaking, trivial beyond a certain dimension: a type A of truncation level n can be characterized by the property that all m-cells for $m > n$ with the same source and target are equal. From this intuitive description we can see that the hierarchy is cumulative.

It is customary to also speak of truncation levels -2 and -1, called *contractible types* and *mere propositions* respectively:

Definition 3. *A type $A : \mathcal{U}_i$ is called* contractible *if there exists a point $a : A$ such that any other point $x : A$ is equal to a:*

$$\mathsf{is\text{-}contr}(A) :\equiv \Sigma_{a:A}\Pi_{x:A}(a =_A x)$$

A type $A : \mathcal{U}_i$ is called a mere proposition *if all its inhabitants are equal:*

$$\mathsf{is\text{-}prop}(A) :\equiv \Pi_{x,y:A}(x =_A y)$$

Thus, a contractible type can be seen as having exactly one inhabitant, up to equality; a mere proposition can be seen as having at most one inhabitant, up to equality. Clearly:

Lemma 4. *If A is contractible then A is a mere proposition.*

The existence of a path between any two points implies more than just path-connectedness:

Lemma 5. *If A is a mere proposition, then $x =_A y$ is contractible for any $x, y : A$.*

Thus, contractible types are in a sense the "nicest" possible: any two points are equal up to a *1-cell*, which itself is unique up to a *2-cell*, which itself is unique up to a *3-cell*, and so on. Mere propositions are the "nicest" ones after contractible spaces. We can now easily show:

Corollary 6. *For any A,* $\mathsf{is\text{-}contr}(A)$ *and* $\mathsf{is\text{-}prop}(A)$ *are mere propositions.*

2.4 Equivalences

A crucial concept in HoTT is that of an equivalence between types.

Definition 7. *A map $f : A \to B$ is called an* equivalence *if it has both a left and a right inverse:*

$$\mathsf{iseq}(f) :\equiv \big(\Sigma_{g:B\to A}(g \circ f \sim \mathsf{id}_A)\big) \times \big(\Sigma_{h:B\to A}(f \circ h \sim \mathsf{id}_B)\big)$$

We define

$$(A \simeq B) :\equiv \Sigma_{f:A\to B}\mathsf{iseq}(f)$$

and call A and B equivalent *if the above type is inhabited.*

Unsurprisingly, we can prove that A and B are equivalent by constructing functions going back and forth, which compose to identity on both sides[1]; this is also a necessary condition.

Lemma 8. *Two types A and B are equivalent if and only if there exist functions $f : A \to B$ and $g : B \to A$ such that $g \circ f \sim \mathsf{id}_A$ and $f \circ g \sim \mathsf{id}_B$.*

We will refer to such functions f and g as forming a *quasi-equivalence* and say that f and g are *quasi-inverses* of each other. From this we can easily show:

Lemma 9. *Equivalence of types is an equivalence relation.*

We call A and B *logically equivalent* if there are exist functions $f : A \to B$, $g : B \to A$. It is immediate that if both types are mere propositions then logical equivalence implies $A \simeq B$. For example:

Corollary 10. *For any A,* $\mathsf{is\text{-}contr}(A) \simeq (A \times \mathsf{is\text{-}prop}(A))$.

Many "diagram-like" operations on paths turn out to be equivalences. For instance:

- For any $u : a =_X b$, $v : b =_X d$, $w : a =_X c$, $z : c =_X d$ there is a map

$$\mathbf{I}^1_\square : (u = w \cdot z \cdot v^{-1}) \to (u \cdot v = w \cdot z)$$

  defined in the obvious way by induction on v and w. This map is an equivalence and we denote its quasi-inverse by $\mathbf{I}^{-1}_\square$.

- For any $u : a =_X b$, $v : b =_X d$, $w : a =_X c$, $z : c =_X d$ there is a map

$$\mathbf{I}^2_\square : (u = w \cdot z \cdot v^{-1}) \to (w^{-1} \cdot u = z \cdot v^{-1})$$

  defined in the obvious way by induction on v and w. This map is an equivalence and we denote its quasi-inverse by $\mathbf{I}^{-2}_\square$.

2.5 Structure of path types

Let us first consider the product type $A \times B$. We would like for two pairs $c, d : A \times B$ to be equal precisely when their first and second projections are equal. By path induction we can easily construct a function

$$^{=}\mathsf{E}^{\times}_{c,d} : (c = d) \to (\pi_1(c) = \pi_1(d)) \times (\pi_2(c) = \pi_2(d))$$

We can show:

Lemma 11. *The map $^{=}\mathsf{E}^{\times}_{c,d}$ is an equivalence for any $c, d : A \times B$.*

[1]Although the type of such functions itself is not equivalent to $A \simeq B$, see Chpt. 4 of [22].

We will denote the quasi-inverse of ${}^{=}\mathsf{E}^{\times}_{c,d}$ by ${}^{\times}\mathsf{E}^{=}_{c,d}$. For brevity we will often omit the subscripts.

We have a similar correspondence for dependent pairs; however, the second projections of $c, d : \Sigma_{x:A} B(x)$ now lie in different fibers of B and we employ (covariant) transport. By path induction we can define a map

$$ {}^{=}\mathsf{E}^{\Sigma}_{c,d} : (c = d) \to \Sigma_{(p:\pi_1(c)=\pi_1(d))}(p^B_*(\pi_2(c)) = \pi_2(d)) $$

Lemma 12. *The map* ${}^{=}\mathsf{E}^{\Sigma}_{c,d}$ *is an equivalence for any* $c, d : \Sigma_{x:A} B(x)$.

We will denote the quasi-inverse of ${}^{=}\mathsf{E}^{\Sigma}_{c,d}$ by ${}^{\Sigma}\mathsf{E}^{=}_{c,d}$. We also have an analogous correspondence using a contravariant transport.

We would like for two types $A, B : \mathcal{U}_i$ to be equal precisely when they are equivalent. As before, we can easily obtain a function

$$ {}^{=}\mathsf{E}^{\simeq}_{A,B} : (A = B) \to (A \simeq B) $$

The univalence axiom now states that this map is an equivalence:

Axiom 1 (Univalence)**.** *The map* ${}^{=}\mathsf{E}^{\simeq}_{A,B}$ *is an equivalence for any* $A, B : \mathcal{U}_i$.

We will denote the quasi-inverse of ${}^{=}\mathsf{E}^{\simeq}_{A,B}$ by ${}^{\simeq}\mathsf{E}^{=}_{A,B}$.

It follows from univalence that *equivalent types are equal* and hence they satisfy the same properties:

Lemma 13. *For any type family* $P : \mathcal{U}_i \to \mathcal{U}_j$*, and types* $A, B : \mathcal{U}_i$ *with* $A \simeq B$*, we have that* $P(A) \simeq P(B)$*. Thus in particular,* $P(A)$ *is inhabited precisely when* $P(B)$ *is.*

Finally, two functions $f, g : \Pi_{x:A} B(x)$ should be equal precisely when there exists a homotopy between them. Constructing a map

$$ {}^{=}\mathsf{E}^{\Pi}_{f,g} : (f = g) \to (f \sim g) $$

is easy. Showing that this map is an equivalence (or even constructing a map in the opposite direction) is much harder, and is in fact among the chief consequences of univalence:

Lemma 14. *The map* ${}^{=}\mathsf{E}^{\Pi}_{f,g}$ *is an equivalence for any* $f, g : \Pi_{x:A} B(x)$.

Proof. See Chpt. 4.9 of [22]. □

We will denote the quasi-inverse of ${}^{=}\mathsf{E}^{\Pi}_{f,g}$ by ${}^{\Pi}\mathsf{E}^{=}_{f,g}$.

3 Higher Inductive Types

An inductive type X can be understood as being *freely generated* by a collection of constructors: in the familiar case of natural numbers, we have the two constructors for zero and successor. The property of being freely generated can be stated as an induction principle: in order to show that a property $P : \mathbb{N} \to \mathcal{U}_i$ holds for all $n : \mathbb{N}$, it suffices to show that it holds for zero and is preserved by the successor operation. As a special case, we get the recursion principle: in order to define a map $f : \mathbb{N} \to C$, is suffices to determine its value at zero and its behavior with respect to successor.

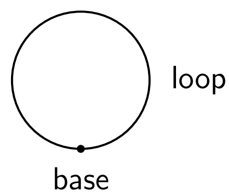
Higher inductive types generalize ordinary inductive types by allowing constructors involving *path spaces of* X rather than just X itself, as the next example shows.

3.1 The circle

The unit circle $\mathbf{S^1}$ is represented as an inductive type $\mathbb{S} : \mathcal{U}_0$ with two constructors [12]:

$$\begin{aligned}&\mathsf{base} : \mathbb{S}\\&\mathsf{loop} : \mathsf{base} =_{\mathbb{S}} \mathsf{base}\end{aligned}$$

pictured as



This in particular means that we have further paths, such as $\mathsf{loop}^{-1} \cdot \mathsf{loop} \cdot \mathsf{loop} \cdot \mathsf{refl}_{\mathbb{S}}(\mathsf{base})$ (which is equal to loop).

We can reason about the circle using the principle of *circle recursion*, also called *simple elimination* for $\mathbb{S}$, which tells us that in order to construct a function out of $\mathbb{S}$ into a type C, it suffices to supply a point $c : C$ and a loop $s : c =_C c$.

$$\frac{C : \mathcal{U}_i \qquad c : C \qquad s : c =_C c}{\mathsf{rec}_{\mathbb{S}}(C, c, s) : \mathbb{S} \to C}$$

Furthermore, the recursor has the expected behavior on the 0-cell constructor base (we omit the premises):

$$\mathsf{rec}_{\mathbb{S}}(C, c, s)(\mathsf{base}) \equiv c : C$$

We also have a computation rule for the 1-cell constructor loop:

$$\mathsf{ap}_{\mathsf{rec}_{\mathbb{S}}(C,c,s)}(\mathsf{loop}) =_{\mathsf{Id}_C(c,c)} s$$

This rule type-checks by virtue of the previous one. We note that in order to record the effect of the recursor on the path loop, we use the "action-on-paths" construct ap. Since this is a derived notion rather than a primitive one, we state the rule as a propositional rather than definitional equality.

We also have the more general principle of *circle induction*, also called *dependent elimination* for $\mathbb{S}$, which subsumes recursion. Instead of a type $C : \mathcal{U}_i$ we now have a type family $E : \mathbb{S} \to \mathcal{U}_i$. Where previously we required a point $c : C$, we now need a point $e : E(\mathsf{base})$. Finally, an obvious generalization of needing a loop $s : c =_C c$ would be to ask for a loop $d : e =_{E(\mathsf{base})} e$. However, this would be incorrect: once we have our desired inductor of type $\Pi_{x:\mathbb{S}} E(x)$, its effect on loop is not a loop at e in the fiber $E(\mathsf{base})$ but a path from $\mathsf{loop}_*^E(e)$ to e in $E(\mathsf{base})$ (or its contravariant version). The induction principle thus takes the following form:

$$\frac{E : \mathbb{S} \to \mathcal{U}_i \qquad e : E(\mathsf{base}) \qquad d : \mathsf{loop}_*^E(e) =_{E(\mathsf{base})} e}{\mathsf{ind}_{\mathbb{S}}(E, e, d) : \Pi_{x:\mathbb{S}} E(x)}$$

We have the associated computation rules:

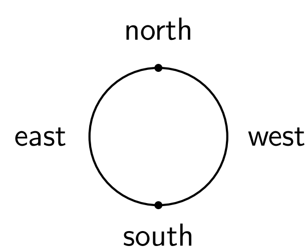
$$\mathsf{ind}_{\mathbb{S}}(E, e, d)(\mathsf{base}) \equiv e : E(\mathsf{base})$$
$$\mathsf{dap}_{\mathsf{ind}_{\mathbb{S}}(E,e,d)}(\mathsf{loop}) =_{\mathsf{Id}_{E(\mathsf{base})}(\mathsf{loop}_*^E(e), e)} d$$

3.2 The circle, round two

We could have alternatively represented the circle as an inductive type $\mathbb{S}_a : \mathcal{U}_0$ with four constructors:

$$\begin{aligned} \mathsf{north} &: \mathbb{S}_a \\ \mathsf{south} &: \mathbb{S}_a \\ \mathsf{east} &: \mathsf{north} =_{\mathbb{S}_a} \mathsf{south} \\ \mathsf{west} &: \mathsf{north} =_{\mathbb{S}_a} \mathsf{south} \end{aligned}$$

pictured as



We now have the recursion principle

$$\frac{C : \mathcal{U}_i \qquad c : C \qquad d : C \qquad p : c =_C d \qquad q : c =_C d}{\mathsf{rec}_{\mathbb{S}_a}(C, c, d, p, q) : \mathbb{S}_a \to C}$$

with the computation rules

$$\mathsf{rec}_{\mathbb{S}_a}(C, c, d, p, q, \mathsf{north}) \equiv c : C$$
$$\mathsf{rec}_{\mathbb{S}_a}(C, c, d, p, q, \mathsf{south}) \equiv d : C$$

and

$$\mathsf{ap}_{\mathsf{rec}_{\mathbb{S}_a}(C,c,d,p,q)}(\mathsf{east}) = p$$
$$\mathsf{ap}_{\mathsf{rec}_{\mathbb{S}_a}(C,c,d,p,q)}(\mathsf{west}) = q$$

The corresponding induction principle is

$$\frac{E : \mathbb{S}_a \to \mathcal{U}_i \qquad u : E(\mathsf{north}) \qquad v : E(\mathsf{south}) \qquad \mu : \mathsf{east}_*^E(u) =_{E(\mathsf{south})} v \qquad \nu : \mathsf{west}_*^E(u) =_{E(\mathsf{south})} v}{\mathsf{ind}_{\mathbb{S}_a}(E, u, v, \mu, \nu) : \Pi_{x:\mathbb{S}_a} E(x)}$$

with the associated computation rules

$$\mathsf{ind}_{\mathbb{S}_a}(E, u, v, \mu, \nu, \mathsf{north}) \equiv u : E(\mathsf{north})$$
$$\mathsf{ind}_{\mathbb{S}_a}(E, u, v, \mu, \nu, \mathsf{south}) \equiv v : E(\mathsf{south})$$

and

$$\mathsf{dap}_{\mathsf{ind}_{\mathbb{S}_a}(E,u,v,\mu,\nu)}(\mathsf{east}) = \mu$$
$$\mathsf{dap}_{\mathsf{ind}_{\mathbb{S}_a}(E,u,v,\mu,\nu)}(\mathsf{west}) = \nu$$

As expected, the two circle types are equivalent:

Lemma 15. *We have* $\mathbb{S} \simeq \mathbb{S}_a$.

Proof sketch. From left to right, map base to north and loop to $\mathsf{east} \cdot \mathsf{west}^{-1}$. From right to left, map both north and south to base, east to loop, and west to $\mathsf{refl}_{\mathbb{S}}(\mathsf{base})$. Using the respective induction principles, show that these two mappings compose to identity on both sides and apply Lem. 8. □

3.3 Computation laws, revisited

By Lem. 15 the types $\mathbb{S}$ and $\mathbb{S}_a$ are equivalent and hence satisfy the same properties (see Lem. 13). We would thus expect the induction principle for $\mathbb{S}$ to carry over to $\mathbb{S}_a$, and vice versa. Indeed, with a little effort we can show the former:

Lemma 16. *The type* $\mathbb{S}_a$ *satisfies the induction and computation laws for* $\mathbb{S}$*, with* north *acting as the constructor* base *and* $\mathsf{east} \cdot \mathsf{west}^{-1}$ *acting as the constructor* loop.

In the other direction, though, we hit a snag - the only obvious choice we have is to define both points north and south to be base, one of the paths west and east to be loop, and the other one the identity path at base. This, however, does not give us the desired induction principle: unless the two given points $u : E(\mathsf{base})$ and $v : E(\mathsf{base})$ happen to be definitionally equal, we will not be able to map base to both of them, as required by the computation rules.

This poses more than just a conceptual problem - in mathematics, we often have several possible definitions of a given notion, all of which are interchangeable from the point of view of a "user". Having two definitions of a circle which are not (known to be) interchangeable, however, can be problematic: any theorem we establish about or by appealing to $\mathbb{S}_a$ might no longer hold - or even type-check! - when using $\mathbb{S}$ instead. To see this, take the second computation law for $\mathbb{S}_a$, $\mathsf{dap}_{\mathsf{ind}_{\mathbb{S}_a}(E,u,v,\mu,\nu)}(\mathsf{west}) = \nu$. If we attempt to "implement" $\mathbb{S}_a$ using the circle $\mathbb{S}$ instead - by taking $\mathsf{north}, \mathsf{south} :\equiv \mathsf{base}$, $\mathsf{east} :\equiv \mathsf{loop}$, $\mathsf{west} :\equiv \mathsf{refl}_{\mathbb{S}}(\mathsf{base})$ as in the proof of Lem. 15 - the computation law is no longer well-typed since the left-hand side reduces to a reflexivity path whereas the right hand side is a path from u to v.

This is one of the motivations for considering inductive types with *propositional* computation behavior: we now want to investigate types which "act like the circle" up to propositional equality. In the case of $\mathbb{S}$, such a type $C : \mathcal{U}_i$ should come with a point $b : C$ and loop $l : c =_C c$. In the case of $\mathbb{S}_a$, such a type should come with two points $n, s : C$ and two paths $e, w : n =_C s$. We can express this more concisely as follows:

Definition 17. *Define the type of* $\mathbb{S}$-algebras *on a universe* $\mathcal{U}_i$ *as*

$$\mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_i} :\equiv \Sigma_{C:\mathcal{U}_i}\Sigma_{b:C}(b = b)$$

Definition 18. *Define the type of* $\mathbb{S}_a$-algebras *on a universe* $\mathcal{U}_i$ *as*

$$\mathbb{S}_a\text{-}\mathsf{Alg}_{\mathcal{U}_i} :\equiv \Sigma_{C:\mathcal{U}_i}\Sigma_{n,s:C}(n = s) \times (n = s)$$

We are now interested in maps between algebras which in a suitable sense preserve the distinguished points and paths, i.e., *algebra homomorphisms*. A homomorphism between two $\mathbb{S}$-algebras (C, c, p) and (D, d, q) should be a function $f : C \to D$ for which we have a path $\beta : f(c) = d$. Furthermore, f should also appropriately relate p and q. To figure out what this means, we observe that if we map p along f, we obtain a path $\mathsf{ap}_f(p) : f(c) = f(c)$. Each of the (identical) endpoints is equal to d, via the path β. Thus, we now have another path $\beta^{-1} \cdot \mathsf{ap}_f(p) \cdot \beta : d = d$. It is reasonable to require that this path be equal to q, i.e., that the following diagram commutes:

$$\begin{array}{ccc} f(c) & \overset{\mathsf{ap}_f(p)}{\overline{\qquad\quad}} & f(c) \\ \beta \,\big| & & \big|\, \beta \\ d & \underset{q}{\overline{\qquad\quad}} & d \end{array}$$

Likewise, a homomorphism between two $\mathbb{S}_a$-algebras (C, a, b, p, q) and (D, c, d, r, s) should be a function $f : C \to D$ for which we have paths $\beta : f(a) = c$, $\gamma : f(b) = d$ and for which the following diagrams commute:

$$\begin{array}{ccc} f(a) & \overset{\mathsf{ap}_f(p)}{\overline{\qquad\quad}} & f(b) \\ \beta \,\big| & & \big|\, \gamma \\ c & \underset{r}{\overline{\qquad\quad}} & d \end{array} \qquad \begin{array}{ccc} f(a) & \overset{\mathsf{ap}_f(q)}{\overline{\qquad\quad}} & f(b) \\ \beta \,\big| & & \big|\, \gamma \\ c & \underset{s}{\overline{\qquad\quad}} & d \end{array}$$

We can express this as follows:

Definition 19. *For algebras* $\mathcal{X} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$, $\mathcal{Y} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_j}$, *define the type of* $\mathbb{S}$-homomorphisms *from* $\mathcal{X}$ *to* $\mathcal{Y}$ *by*

$$\mathbb{S}\text{-}\mathsf{Hom}\ (C, c, p)\ (D, d, q) :\equiv \Sigma_{f:C\to D}\Sigma_{\beta:f(c)=d}\big(\mathsf{ap}_f(p) = \beta \cdot q \cdot \beta^{-1}\big)$$

Definition 20. *For algebras* $\mathcal{X} : \mathbb{S}_a\text{-}\mathsf{Alg}_{\mathcal{U}_i}$, $\mathcal{Y} : \mathbb{S}_a\text{-}\mathsf{Alg}_{\mathcal{U}_j}$, *define the type of* $\mathbb{S}_a$-homomorphisms *from* $\mathcal{X}$ *to* $\mathcal{Y}$ *by*

$$\mathbb{S}_a\text{-}\mathsf{Hom}\ (C, a, b, p, q)\ (D, c, d, r, s) :\equiv \Sigma_{f:C\to D}\Sigma_{\beta:f(a)=c}\Sigma_{\gamma:f(b)=d}$$
$$\big(\mathsf{ap}_f(p) = \beta \cdot r \cdot \gamma^{-1}\big) \times \big(\mathsf{ap}_f(q) = \beta \cdot s \cdot \gamma^{-1}\big)$$

We note that to be able to form the type of homomorphisms as we just did, it is crucial to have the computation laws stated propositionally. The recursion principle now becomes a property internal to the type theory and can be expressed compactly as saying that there is a homomorphism into any other algebra $\mathcal{Y}$:

Definition 21. *An algebra* $\mathcal{X} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ satisfies the $\mathbb{S}$-recursion principle *on a universe* $\mathcal{U}_j$ *if for any algebra* $\mathcal{Y} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_j}$ *there exists a homomorphism from* $\mathcal{X}$ *to* $\mathcal{Y}$*:*

$$\mathsf{has}\text{-}\mathbb{S}\text{-}\mathsf{rec}_{\mathcal{U}_j}(\mathcal{X}) :\equiv \big(\Pi\mathcal{Y} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_j}\big)\ \mathbb{S}\text{-}\mathsf{Hom}\ \mathcal{X}\ \mathcal{Y}$$

Definition 22. *An algebra* $\mathcal{X} : \mathbb{S}_a\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ satisfies the $\mathbb{S}_a$-recursion principle *on a universe* $\mathcal{U}_j$ *if for any algebra* $\mathcal{Y} : \mathbb{S}_a\text{-}\mathsf{Alg}_{\mathcal{U}_j}$ *there exists a homomorphism from* $\mathcal{X}$ *to* $\mathcal{Y}$*:*

$$\mathsf{has}\text{-}\mathbb{S}_a\text{-}\mathsf{rec}_{\mathcal{U}_j}(\mathcal{X}) :\equiv \big(\Pi\mathcal{Y} : \mathbb{S}_a\text{-}\mathsf{Alg}_{\mathcal{U}_j}\big)\ \mathbb{S}_a\text{-}\mathsf{Hom}\ \mathcal{X}\ \mathcal{Y}$$

To express the induction principle in a similar fashion, we first need to introduce dependent or *fibered* versions of algebras and algebra homomorphisms:

Definition 23. *Define the type of* fibered $\mathbb{S}$-algebras *on a universe* $\mathcal{U}_j$ *over an algebra* $\mathcal{X} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ *by*

$$\mathbb{S}\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_j}\ (C, c, p) :\equiv \Sigma_{E:C\to\mathcal{U}_j}\Sigma_{e:E(c)}\big(p_*^E(e) = e\big)$$

Definition 24. *Define the type of* fibered $\mathbb{S}_a$-algebras *on a universe* $\mathcal{U}_j$ *over an algebra* $\mathcal{X}$: $\mathbb{S}_a\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ *by*

$$\mathbb{S}_a\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_j}\ (C, c, d, p, q) :\equiv \Sigma_{E:C\to\mathcal{U}_j}\Sigma_{u:E(c)}\Sigma_{v:E(d)}\big(p_*^E(u) = v\big) \times \big(q_*^E(u) = v\big)$$

Definition 25. *For algebras* $\mathcal{X} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$, $\mathcal{Y} : \mathbb{S}\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_j}\ \mathcal{X}$, *define the type of* fibered $\mathbb{S}$-homomorphisms *from* $\mathcal{X}$ *to* $\mathcal{Y}$ *by*

$$\mathbb{S}\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\ (C, c, p)\ (E, e, q) :\equiv \Sigma_{f:(\Pi x:C)E(x)}\Sigma_{\beta:f(c)=e}\big(\mathsf{dap}_f(p) = \mathsf{ap}_{p_*^E}(\beta) \cdot q \cdot \beta^{-1}\big)$$

Pictorially, the last component witnesses the commuting diagram

$$\begin{array}{ccc} p^E_*(f(c)) & \overset{\mathsf{dap}_f(p)}{\rule[0.5ex]{4em}{0.4pt}} & f(c) \\ {\scriptstyle \mathsf{ap}_{p^E_*}(\beta)}\Big| & & \Big|{\scriptstyle \beta} \\ p^E_*(e) & \underset{q}{\rule[0.5ex]{4em}{0.4pt}} & e \end{array}$$

Definition 26. *For algebras* $\mathcal{X} : \mathbb{S}_a\text{-}\mathsf{Alg}_{\mathcal{U}_i}$, $\mathcal{Y} : \mathbb{S}_a\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_j}\ \mathcal{X}$*, define the type of* fibered $\mathbb{S}_a$-homomorphisms *from* $\mathcal{X}$ *to* $\mathcal{Y}$ *by*

$$\mathbb{S}_a\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\ (C, a, b, p, q)\ (D, c, d, r, s) :\equiv \Sigma_{f:(\Pi x:C)E(x)}\Sigma_{\beta:f(a)=c}\Sigma_{\gamma:f(b)=d}$$
$$\big(\mathsf{dap}_f(p) = \mathsf{ap}_{p^E_*}(\beta) \cdot r \cdot \gamma^{-1}\big) \times \big(\mathsf{dap}_f(q) = \mathsf{ap}_{q^E_*}(\beta) \cdot s \cdot \gamma^{-1}\big)$$

Pictorially, the last two components witness the commuting diagrams

$$\begin{array}{ccc} p^E_*(f(a)) & \overset{\mathsf{dap}_f(p)}{\rule[0.5ex]{4em}{0.4pt}} & f(b) \\ {\scriptstyle \mathsf{ap}_{p^E_*}(\beta)}\Big| & & \Big|{\scriptstyle \gamma} \\ p^E_*(c) & \underset{r}{\rule[0.5ex]{4em}{0.4pt}} & d \end{array} \qquad\qquad \begin{array}{ccc} p^E_*(f(a)) & \overset{\mathsf{dap}_f(q)}{\rule[0.5ex]{4em}{0.4pt}} & f(b) \\ {\scriptstyle \mathsf{ap}_{p^E_*}(\beta)}\Big| & & \Big|{\scriptstyle \gamma} \\ p^E_*(c) & \underset{s}{\rule[0.5ex]{4em}{0.4pt}} & d \end{array}$$

The induction principle can now be expressed as saying that there is a fibered homomorphism into any fibered algebra $\mathcal{Y}$:

Definition 27. *An algebra* $\mathcal{X} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ satisfies the $\mathbb{S}$-induction principle *on a universe* $\mathcal{U}_j$ *if for any fibered algebra* $\mathcal{Y} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_j}\ \mathcal{X}$ *there exists a fibered homomorphism from* $\mathcal{X}$ *to* $\mathcal{Y}$*:*

$$\mathsf{has}\text{-}\mathbb{S}\text{-}\mathsf{ind}_{\mathcal{U}_j}(\mathcal{X}) :\equiv \big(\Pi\mathcal{Y} : \mathbb{S}\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_j}\big)\ \mathbb{S}\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\ \mathcal{X}\ \mathcal{Y}$$

Definition 28. *An algebra* $\mathcal{X} : \mathbb{S}_a\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ satisfies the $\mathbb{S}_a$-induction principle *on universe* $\mathcal{U}_j$ *if for any fibered algebra* $\mathcal{Y} : \mathbb{S}_a\text{-}\mathsf{Alg}_{\mathcal{U}_j}\ \mathcal{X}$ *there exists a fibered homomorphism from* $\mathcal{X}$ *to* $\mathcal{Y}$*:*

$$\mathsf{has}\text{-}\mathbb{S}_a\text{-}\mathsf{ind}_{\mathcal{U}_j}(\mathcal{X}) :\equiv \big(\Pi\mathcal{Y} : \mathbb{S}_a\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_j}\big)\ \mathbb{S}_a\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\ \mathcal{X}\ \mathcal{Y}$$

3.4 Relating the two circles

We first note that the notions of $\mathbb{S}$-algebras and $\mathbb{S}_a$-algebras are in fact the same:

Lemma 29. *We have a function*

$$\mathbb{S}\text{-}\mathsf{to}\text{-}\mathbb{S}_a\text{-}\mathsf{Alg}_{\mathcal{U}_i} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_i} \to \mathbb{S}_a\text{-}\mathsf{Alg}_{\mathcal{U}_i}$$

which is an equivalence.

Proof. Define the equivalence between $\mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ and $\mathbb{S}_a\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ by the quasi-inverses

$$\begin{aligned}(C, c, p) &\mapsto (C, c, c, p, \mathsf{refl}(c))\\ (C, a, b, p, q) &\mapsto (C, a, p \cdot q^{-1})\end{aligned}$$

□

Next, we note that the notions of fibered $\mathbb{S}$-algebras and fibered $\mathbb{S}_a$-algebras are the same, in the following sense:

Lemma 30. *For any algebra $\mathcal{X} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ we have a function*

$$\mathbb{S}\text{-to-}\mathbb{S}_a\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_i}(\mathcal{X}) : \mathbb{S}\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_i}\ \mathcal{X} \to \mathbb{S}_a\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_i}\ \big(\mathbb{S}\text{-to-}\mathbb{S}_a\text{-}\mathsf{Alg}_{\mathcal{U}_i}\ \mathcal{X}\big)$$

which is an equivalence.

Proof. Fix algebra $(C, c, p) : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$. Define the equivalence between $\mathbb{S}\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_i}\ (C, c, p)$ and $\mathbb{S}_a\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_i}\ (C, c, c, p, \mathsf{refl}(c))$ by the quasi-inverses

$$\begin{aligned}(E, e, q) &\mapsto (E, e, e, q, \mathsf{refl}(e))\\ (E, a, b, r, s) &\mapsto (E, a, r \cdot s^{-1})\end{aligned}$$

□

The notions of $\mathbb{S}$-homomorphisms and $\mathbb{S}_a$-homomorphisms also coincide:

Lemma 31. *For any algebras $\mathcal{X} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$, $\mathcal{Y} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_j}$ we have*

$$\mathbb{S}\text{-}\mathsf{Hom}\ \mathcal{X}\ \mathcal{Y} \simeq \mathbb{S}_a\text{-}\mathsf{Hom}\ \big(\mathbb{S}\text{-to-}\mathbb{S}_a\text{-}\mathsf{Alg}_{\mathcal{U}_i}\ \mathcal{X}\big)\ \big(\mathbb{S}\text{-to-}\mathbb{S}_a\text{-}\mathsf{Alg}_{\mathcal{U}_i}\ \mathcal{Y}\big)$$

Finally, the respective fibered versions of $\mathbb{S}$-homomorphisms and $\mathbb{S}_a$-homomorphisms coincide:

Lemma 32. *For any algebras $\mathcal{X} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$, $\mathcal{Y} : \mathbb{S}\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_j}\ \mathcal{X}$ we have*

$$\mathbb{S}\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\ \mathcal{X}\ \mathcal{Y} \simeq \mathbb{S}_a\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\ \big(\mathbb{S}\text{-to-}\mathbb{S}_a\text{-}\mathsf{Alg}_{\mathcal{U}_i}\ \mathcal{X}\big)\ \big(\mathbb{S}\text{-to-}\mathbb{S}_a\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_i}(\mathcal{X})\ \mathcal{Y}\big)$$

We can now show that $\mathbb{S}$-recursion is the same as $\mathbb{S}_a$-recursion, and likewise for induction:

Lemma 33. *For any $\mathcal{X} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ we have*

$$\begin{aligned}\mathsf{has}\text{-}\mathbb{S}\text{-}\mathsf{rec}_{\mathcal{U}_j}(\mathcal{X}) &\simeq \mathsf{has}\text{-}\mathbb{S}_a\text{-}\mathsf{rec}_{\mathcal{U}_j}(\mathbb{S}\text{-to-}\mathbb{S}_a\text{-}\mathsf{Alg}_{\mathcal{U}_i}(\mathcal{X}))\\ \mathsf{has}\text{-}\mathbb{S}\text{-}\mathsf{ind}_{\mathcal{U}_j}(\mathcal{X}) &\simeq \mathsf{has}\text{-}\mathbb{S}_a\text{-}\mathsf{ind}_{\mathcal{U}_j}(\mathbb{S}\text{-to-}\mathbb{S}_a\text{-}\mathsf{Alg}_{\mathcal{U}_i}(\mathcal{X}))\end{aligned}$$

Corollary 34. *The $\mathbb{S}_a$-algebra $(\mathbb{S}, \mathsf{base}, \mathsf{base}, \mathsf{loop}, \mathsf{refl}_{\mathbb{S}}(\mathsf{base}))$ satisfies the $\mathbb{S}_a$-induction principle on any universe $\mathcal{U}_j$.*

Corollary 35. *The $\mathbb{S}$-algebra $(\mathbb{S}_a, \mathsf{north}, \mathsf{east} \cdot \mathsf{west}^{-1})$ satisfies the $\mathbb{S}$-induction principle on any universe $\mathcal{U}_j$.*

4 Propositional Truncation

Another example of a higher inductive type is the *propositional truncation* $||A|| : \mathcal{U}_i$ of a type $A : \mathcal{U}_i$, investigated in [1] in an extensional setting under the name *bracket types*. Intuitively, $||A||$ represents the "squashing" of A which makes all the elements in A equal. The need for such a type arises when we wish to hide information: having a term $a : A$ is very different from having a $b : ||A||$. In the latter case, we know that the *provable failure* of A to be inhabited, that is, a term of type $A \to \mathbf{0}$, would lead to a contradiction. However, we do not have a generic way of constructing an inhabitant of A.

4.1 The type $||A||$

We define $||A||$ as the higher inductive type generated by a constructor $|\cdot|$, which projects a given element of A down to $||A||$, and a truncation constructor, which states that $||A||$ is indeed a mere proposition[2]:

$$|\cdot| : A \to ||A||$$
$$\mathsf{sq} : \Pi_{x,y:||A||}(x =_{||A||} y)$$

As usual, the recursion principle states that given a structure of the same form, we have a function out of $||A||$ which preserves the constructors:

$$\frac{C : \mathcal{U}_j \qquad c : A \to C \qquad s : \Pi_{x,y:C}(x =_C y)}{\mathsf{rec}_{||A||}(C, c, s) : ||A|| \to C}$$

where for each $a : A$ we have

$$\mathsf{rec}_{||A||}(C, c, s, |a|) \equiv c(a) : C$$

and for each $k, l : ||A||$ we have

$$\mathsf{ap}_{\mathsf{rec}_{||A||}(C,c,s)}(\mathsf{sq}(k, l)) = s\big(\mathsf{rec}_{||A||}(C, c, s, k), \mathsf{rec}_{||A||}(C, c, s, l)\big)$$

We note that we are only able to eliminate into types which are themselves mere propositions. This together with Lem. 5 implies that the second computation law always holds. We have included it nonetheless to illustrate the general pattern.

To state the induction principle, we need to suitably generalize the last hypothesis. As before, we note that once the desired map $f : \Pi_{x:||A||}E(x)$ is constructed, it will give us a path from $\mathsf{sq}(k, l)^E_*(f(k))$ to $f(l)$ in $E(l)$ for any $k, l : ||A||$. Hence, E should already come equipped with such a family of paths - except, of course, we have no way of referring to $f(k)$ and $f(l)$ before f is constructed. Thus, we simply require that such a path exists for *all* points $u : E(k)$ and $v : E(l)$:

$$\frac{E : ||A|| \to \mathcal{U}_j \qquad e : \Pi_{a:A}E(|a|) \qquad q : \Pi_{x,y:||A||}\Pi_{u:E(x)}\Pi_{v:E(y)}\big(\mathsf{sq}(x, y)^E_*(u) =_{E(y)} v\big)}{\mathsf{ind}_{||A||}(E, e, q) : \Pi_{x:||A||}E(x)}$$

[2]Hence the name *propositional* truncation; see Chpt. 6 of [22] for other kinds of truncation.

where for each $a : A$ we have

$$\mathsf{ind}_{||A||}(E, e, q, |a|) \equiv e(a) : E(|a|)$$

and for each $k, l : ||A||$ we have

$$\mathsf{dap}_{\mathsf{ind}_{||A||}(E,e,q)}(\mathsf{sq}(k, l)) = q\big(k, l, \mathsf{ind}_{||A||}(E, e, q, k), \mathsf{ind}_{||A||}(E, e, q, l)\big)$$

The second rule again turns out to always hold, as we will see shortly.

4.2 Propositional computation laws

We point out that the truncation $||A||$ as defined has its share of unexpected behavior. For instance, as the type $\mathbb{N}$ of natural numbers is inhabited, it follows that $||\mathbb{N}|| = \mathbf{1}$. It is not obvious, however, how to turn $\mathbf{1}$ itself into a truncation of $\mathbb{N}$, since the first computation law ought to hold *definitionally*. More surprising yet is the observation by N. Kraus in [9] that there exists a map f such that $f \circ |\cdot| \equiv \mathsf{id}_{\mathbb{N}}$; this is another somewhat strange side effect of the definitional computation law for $|\cdot|$.

In light of these issues, we follow our methodology from the previous section and investigate types which "act like the type $||A||$" up to propositional equality. We have the following:

Definition 36. *Define the type of* $||A||$-algebras *on a universe* $\mathcal{U}_j$ *as*

$$||A||\text{-}\mathsf{Alg}_{\mathcal{U}_j} :\equiv \Sigma_{C:\mathcal{U}_j}(A \to C) \times \mathsf{is\text{-}prop}(A)$$

A natural definition of an algebra homomorphism between two $||A||$-algebras (C, c, p) and (D, d, q) is a map $f : C \to D$ together with path families

$$\beta : \Pi_{a:A}\big(f(c(a)) = d(a)\big)$$
$$\gamma : \Pi_{x,y:C}\big(\mathsf{ap}_f(p(x, y)) = q(f(x), f(y))\big)$$

However, the type D is a mere proposition. Thus by Lem. 5 and the fact that a family of contractible types is itself contractible, it follows that both of the above types are equivalent to $\mathbf{1}$. Hence, we have the simple definition:

Definition 37. *Given algebras* $\mathcal{X} : ||A||\text{-}\mathsf{Alg}_{\mathcal{U}_j}$ *and* $\mathcal{Y} : ||A||\text{-}\mathsf{Alg}_{\mathcal{U}_k}$*, we define the type of* $||A||$-homomorphisms *from* $\mathcal{X}$ *to* $\mathcal{Y}$ *by*

$$||A||\text{-}\mathsf{Hom}\ (C, c, p)\ (D, d, q) :\equiv C \to D$$

As before, the recursion principle states that there is a homomorphism to any other algebra $\mathcal{Y}$:

Definition 38. *An algebra* $\mathcal{X} : ||A||\text{-}\mathsf{Alg}_{\mathcal{U}_j}$ satisfies the recursion principle *on a universe* $\mathcal{U}_k$ *if for any algebra* $\mathcal{Y} : ||A||\text{-}\mathsf{Alg}_{\mathcal{U}_k}$ *there exists a* $||A||$*-homomorphism from* $\mathcal{X}$ *to* $\mathcal{Y}$*:*

$$\mathsf{has}\text{-}||A||\text{-}\mathsf{rec}_{\mathcal{U}_k}(\mathcal{X}) :\equiv \big(\Pi\mathcal{Y} : ||A||\text{-}\mathsf{Alg}_{\mathcal{U}_k}\big)\ ||A||\text{-}\mathsf{Hom}\ \mathcal{X}\ \mathcal{Y}$$

Based on the $||A||$-elimination rule, a natural definition of a fibered algebra over (C, c, p) is a family of types $E : C \to \mathcal{U}_k$ endowed with a function $e : \Pi_{a:A} E(c(a))$ and path family

$$q : \Pi_{x,y:C} \Pi_{u:E(x)} \Pi_{v:E(y)} (p(x,y)_*^E(u) = v)$$

Using the fact that C is a mere proposition, we can show, however, that the above type is equivalent to the condition that E is a family of mere propositions, $\Pi_{x:C}\mathsf{is\text{-}prop}(E(x))$. We can thus define:

Definition 39. *Define the type of fibered $||A||$-algebras on a universe $\mathcal{U}_k$ over $\mathcal{X} : ||A||\text{-}\mathsf{Alg}_{\mathcal{U}_j}$ by*

$$||A||\text{-}\mathsf{Fib\text{-}Alg}_{\mathcal{U}_k}\ (C, c, p) :\equiv \Sigma_{E:C \to \mathcal{U}_k} \big(\Pi_{a:A} E(c(a))\big) \times \big(\Pi_{x:C} \mathsf{is\text{-}prop}(E(x))\big)$$

Analogously to the non-fibered case, a natural definition of a fibered $||A||$-homomorphism from (C, c, p) to (E, e, q) is a function $f : \Pi_{x:C} E(x)$ together with path families

$$\beta : \Pi_{a:A} \big(f(c(a)) = e(a)\big)$$
$$\gamma : \Pi_{x,y:C} \big(\mathsf{dap}_f(p(x,y)) = q(x, y, f(x), f(y))\big)$$

Since E is a family of mere propositions, by exactly the same reasoning we get that both of the above types are equivalent to $\mathbf{1}$. Hence, we can define:

Definition 40. *For algebras $\mathcal{X} : ||A||\text{-}\mathsf{Alg}_{\mathcal{U}_j}$, $\mathcal{Y} : ||A||\text{-}\mathsf{Alg}_{\mathcal{U}_k}\ \mathcal{X}$, define the type of* fibered $||A||$-homomorphisms *from $\mathcal{X}$ to $\mathcal{Y}$ by*

$$||A||\text{-}\mathsf{Fib\text{-}Hom}\ (C, c, p)\ (E, e, q) :\equiv \Pi_{x:C} E(x)$$

As before, the induction principle states that there is a fibered homomorphism to any fibered algebra $\mathcal{Y}$:

Definition 41. *An algebra $\mathcal{X} : ||A||\text{-}\mathsf{Alg}_{\mathcal{U}_j}$* satisfies the $||A||$-induction principle *on a universe $\mathcal{U}_k$ if for any fibered algebra $\mathcal{Y} : ||A||\text{-}\mathsf{Alg}_{\mathcal{U}_k}\ \mathcal{X}$ there exists a fibered homomorphism from $\mathcal{X}$ to $\mathcal{Y}$:*

$$\mathsf{has\text{-}}||A||\mathsf{\text{-}ind}_{\mathcal{U}_k}(\mathcal{X}) :\equiv \big(\Pi\mathcal{Y} : ||A||\text{-}\mathsf{Alg}_{\mathcal{U}_k}\ \mathcal{X}\big)\ ||A||\text{-}\mathsf{Fib\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}$$

Finally, we can show that induction and recursion for $||A||$ are in fact equivalent. We note that since universe levels are cumulative, the technical restriction that $k \geq j$ does not pose a problem.

Theorem 42. *For $A : \mathcal{U}_i$, the following conditions on an algebra $\mathcal{X} : ||A||\text{-}\mathsf{Alg}_{\mathcal{U}_j}$ are equivalent:*

- *$\mathcal{X}$ satisfies the induction principle on the universe $\mathcal{U}_k$*
- *$\mathcal{X}$ satisfies the recursion principle on the universe $\mathcal{U}_k$*

for $k \geq j$. In other words, we have

$$\mathsf{has\text{-}}||A||\mathsf{\text{-}ind}_{\mathcal{U}_k}(\mathcal{X}) \simeq \mathsf{has\text{-}}||A||\mathsf{\text{-}rec}_{\mathcal{U}_k}(\mathcal{X})$$

provided $k \geq j$. Moreover, the two types above are mere propositions.

Proof. The fact that the types are mere propositions is clear. The direction from right to left is obvious. For the other direction, fix algebras $(C, c, p) : ||A||\text{-}\mathsf{Alg}_{\mathcal{U}_j}$, $(E, e, q) : ||A||\text{-}\mathsf{Fib\text{-}Alg}_{\mathcal{U}_k}\ (C, c, p)$. The total space $\Sigma_{x:C} E(x) : \mathcal{U}_k$ is a mere proposition, we can thus apply recursion with the projection map $a \mapsto (c(a), e(a))$ to get a function $f : C \to \Sigma_{x:C} E(x)$. A homotopy $\alpha : \mathsf{fst} \circ f \sim \mathsf{id}_C$ exists as C is a mere proposition. Applying second projection and transporting gives us a map $x \mapsto \alpha(x)_*^E(\pi_2 f(x))$ and we are done. ∎

5 W-Suspensions as Homotopy-Initial Algebras

Here we consider a class of higher inductive types which we call W-*suspensions*; informally, they combine Martin-Löf's W-types [17], also known as well-founded trees, with (a generalized form of) suspensions ([22], Chpt. 6.5). Ordinary W-types allow proper induction on the level of points but have no higher-dimensional constructors. Suspensions, on the other hand, only provide vacuous induction on the point level, in the form of two nullary constructors; however, they allow us to specify an arbitrary number of path constructors between these two points. A suitable combination of these two classes of types keeps the phenomenons of induction and higher-dimensionality orthogonal, which gives us a well-behaved elimination principle.

5.1 W-suspensions

Formally, given types $A, C : \mathcal{U}_i$, a type family $B : A \to \mathcal{U}_i$, and functions $\mathtt{l}, \mathtt{r} : C \to A$, the W-suspension $\mathsf{W}(A, B, C, \mathtt{l}, \mathtt{r}) : \mathcal{U}_i$ is the higher inductive type generated by the constructors

$$\begin{aligned}
&\mathsf{sup} : \Pi_{a:A}\big(B(a) \to \mathsf{W}(A, B, C, \mathtt{l}, \mathtt{r})\big) \to \mathsf{W}(A, B, C, \mathtt{l}, \mathtt{r}) \\
&\mathsf{cell} : \Pi_{c:C}\Pi_{t:B(\mathtt{l}\ c)\to\mathsf{W}(A,B,C,\mathtt{l},\mathtt{r})}\Pi_{s:B(\mathtt{r}\ c)\to\mathsf{W}(A,B,C,\mathtt{l},\mathtt{r})}\ \mathsf{sup}(\mathtt{l}\ c, t) = \mathsf{sup}(\mathtt{r}\ c, s)
\end{aligned}$$

From now on we will keep the arguments $A, B, C, \mathtt{l}, \mathtt{r}$ implicit, unless indicated otherwise.

As in the case of ordinary W-types, the type A can be thought of as the type of labels for points and for any $a : A$, the type $B(a)$ represents the arity of the label a, i.e., it is the index type for the arguments of a. Similarly, the type C represents the type of labels for paths between points. For any $c : C$, the terms $\mathtt{l}(c)$ and $\mathtt{r}(c)$ determine the respective labels of the left and right endpoints of the paths labeled by c. As can be read off from the type of the constructor cell, each label $c : C$ determines a family of paths in W, one for each pair of terms $t : B(\mathtt{l}\ c) \to \mathsf{W}$ and $s : B(\mathtt{r}\ c) \to \mathsf{W}$.

An ordinary W-type $\mathsf{W}_{x:A}B(x)$ arises as a W-suspension in the obvious way by taking $A :\equiv A$, $B :\equiv B$, $C :\equiv \mathbf{0}$, and letting both $\mathtt{l}$ and $\mathtt{r}$ be the canonical function from $\mathbf{0}$ into A. We can encode the circle $\mathbb{S}$ by taking $A, C :\equiv \mathbf{1}$, $B :\equiv \lambda_{\_:\mathbf{1}}\mathbf{0}$, $\mathtt{l}, \mathtt{r} :\equiv \lambda_{\_:\mathbf{1}}\star$. The circle $\mathbb{S}_a$ arises when we take $A, C :\equiv \mathbf{2}$, $B :\equiv \lambda_{\_:\mathbf{2}}\mathbf{0}$, $\mathtt{l} :\equiv \lambda_{\_:\mathbf{2}}\top$, $\mathtt{r} :\equiv \lambda_{\_:\mathbf{2}}\bot$. Other types which can be represented in this form include the interval type, suspensions - hence in particular all the higher spheres $\mathbf{S}^n$ - and of course all ordinary inductive types arising as W-types, e.g., natural numbers, lists, and so on. For more detail we refer to Sect. 5.3.

W-suspensions come with the expected recursion principle: Given terms

- $E : \mathcal{U}_j$
- $e : \Pi_{a:A}(B(a) \to E) \to E$
- $q : \Pi_{c:C}\Pi_{u:B(\mathtt{l}\ c)\to E}\Pi_{v:B(\mathtt{r}\ c)\to E}\big(e(\mathtt{l}\ c, u) = e(\mathtt{r}\ c, v)\big)$

there is a recursor $\mathsf{rec}_{\mathsf{W}}(E, e, q) : \mathsf{W} \to E$. The recursor satisfies the computation laws

- $\mathsf{rec}_{\mathsf{W}}(E, e, q, \mathsf{sup}(a, t)) \equiv e\big(a, \mathsf{rec}_{\mathsf{W}}(E, e, q) \circ t\big)$

for any $a : A, t : B(a) \to \mathsf{W}$ and

- $\mathsf{ap}_{\mathsf{rec}_{\mathsf{W}}(E,e,q)}(\mathsf{cell}(c,t,s)) = q\Big(c, \mathsf{rec}_{\mathsf{W}}(E,e,q) \circ t, \mathsf{rec}_{\mathsf{W}}(E,e,q) \circ s\Big)$

for any $c : C, t : B(\mathtt{l}\ c) \to \mathsf{W}, s : B(\mathtt{r}\ c) \to \mathsf{W}$. Similarly, we have an induction principle: Given terms

- $E : \mathsf{W} \to \mathcal{U}_j$
- $e : \Pi_{a:A}\Pi_{t:B(a)\to\mathsf{W}}\big(\Pi_{b:B(a)}E(t\ b)\big) \to E(\mathsf{sup}(a,t))$
- $q : \Pi_{c:C}\Pi_{t:B(\mathtt{l}\ c)\to\mathsf{W}}\Pi_{s:B(\mathtt{r}\ c)\to\mathsf{W}}\Pi_{u:(\Pi b:B(\mathtt{l}\ c))E(t\ b)}\Pi_{v:(\Pi b:B(\mathtt{r}\ c))E(s\ b)}$
  $\big(\mathsf{cell}(c,t,s)_*^E\ e(\mathtt{l}\ c,t,u) = e(\mathtt{r}\ c,s,v)\big)$

there is an inductor $\mathsf{ind}_{\mathsf{W}}(E,e,q) : \Pi_{w:\mathsf{W}}E(w)$. The inductor satisfies the computation laws

- $\mathsf{ind}_{\mathsf{W}}(E,e,q,\mathsf{sup}(a,t)) \equiv e\big(a,t,\mathsf{ind}_{\mathsf{W}}(E,e,q) \circ t\big)$

for any $a : A, t : B(a) \to \mathsf{W}$ and

- $\mathsf{dap}_{\mathsf{ind}_{\mathsf{W}}(E,e,q)}(\mathsf{cell}(c,t,s)) = q\Big(c,t,s,\mathsf{ind}_{\mathsf{W}}(E,e,q) \circ t, \mathsf{ind}_{\mathsf{W}}(E,e,q) \circ s\Big)$

for any $c : C, t : B(\mathtt{l}\ c) \to \mathsf{W}, s : B(\mathtt{r}\ c) \to \mathsf{W}$.

Following the now-familiar pattern, we define W-suspension algebras and homomorphisms, together with their fibered counterparts. For convenience, we first introduce the corresponding notions for ordinary W-types $\mathsf{W}_{x:A}B(x)$, which we refer to as W-trees, taking into account only the 0-constructor sup; we subsequently extend the definitions to the more general case.

Definition 43. *Define the type of* W-tree algebras *on a universe* $\mathcal{U}_j$ *by*

$$\mathsf{W}_0\text{-}\mathsf{Alg}_{\mathcal{U}_j}(A,B) :\equiv \Sigma_{D:\mathcal{U}_j}\Pi_{a:A}(B(a) \to D) \to D$$

Definition 44. *Define a type family over the type* $\mathsf{W}_0\text{-}\mathsf{Alg}_{\mathcal{U}_j}(A,B)$ *by*

$$\mathsf{W}_1\text{-}\mathsf{Alg}\ (D,d) :\equiv \Pi_{c:C}\Pi_{u:B(\mathtt{l}\ c)\to D}\Pi_{v:B(\mathtt{r}\ c)\to D}\big(d(\mathtt{l}\ c,u) = d(\mathtt{r}\ c,v)\big)$$

Definition 45. *Define the type of* W-suspension algebras *on a universe* $\mathcal{U}_j$ *by*

$$\mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_j}(A,B,C,\mathtt{l},\mathtt{r}) :\equiv \Big(\Sigma\mathcal{X}_0 : \mathsf{W}_0\text{-}\mathsf{Alg}_{\mathcal{U}_j}(A,B)\Big)\ \mathsf{W}_1\text{-}\mathsf{Alg}\ \mathcal{X}_0$$

As before, we will keep the parameters $A, B, C, \mathtt{l}, \mathtt{r}$ implicit if no confusion arises in doing so.

Definition 46. *For an algebra* $\mathcal{X}_0 : \mathsf{W}_0\text{-}\mathsf{Alg}_{\mathcal{U}_j}$*, define the type of* fibered W-tree algebras *on a universe* $\mathcal{U}_k$ *over* $\mathcal{X}_0$ *by*

$$\mathsf{W}_0\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_k}\ (D,d) :\equiv \Sigma_{E:D\to\mathcal{U}_k}\Pi_{a:A}\Pi_{t:B(a)\to D}\big(\Pi_{b:B(a)}E(t\ b)\big) \to E(d(a,t))$$

Definition 47. *For an algebra* $\mathcal{X} : \mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_j}$*, define a type family over the type* $\mathsf{W}_0\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_j}\ \pi_1(\mathcal{X})$ *by*

$$\mathsf{W}_1\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}\ (D,d,p)\ (E,e) :\equiv \Pi_{c:C}\Pi_{t:B(\mathtt{l}\ c)\to D}\Pi_{s:B(\mathtt{r}\ c)\to D}$$
$$\Pi_{u:(\Pi b:B(\mathtt{l}\ c))E(t\ b)}\Pi_{v:(\Pi b:B(\mathtt{r}\ c))E(s\ b)}\left(p(c,t,s)_*^E\ e(\mathtt{l}\ c,t,u) = e(\mathtt{r}\ c,s,v)\right)$$

Definition 48. *For an algebra* $\mathcal{X} : \mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_j}$*, define the type of* fibered W-suspension algebras *on a universe* $\mathcal{U}_k$ *over* $\mathcal{X}$ *by*

$$\mathsf{W}\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_k}\ \mathcal{X} :\equiv \left(\Sigma\mathcal{Y}_0 : \mathsf{W}_0\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_j}\ \pi_1(\mathcal{X})\right)\ \mathsf{W}_1\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}\ \mathcal{X}\ \mathcal{Y}_0$$

To express the type of homomorphisms between two W-type or W-suspension algebras, we again need to use propositional instead of definitional equality.

Definition 49. *For algebras* $\mathcal{X}_0 : \mathsf{W}_0\text{-}\mathsf{Alg}_{\mathcal{U}_j}$ *and* $\mathcal{Y}_0 : \mathsf{W}_0\text{-}\mathsf{Alg}_{\mathcal{U}_k}$*, define the type of* W-tree homomorphisms *from* $\mathcal{X}_0$ *to* $\mathcal{Y}_0$ *by*

$$\mathsf{W}_0\text{-}\mathsf{Hom}\ (D,d)\ (E,e) :\equiv \Sigma_{f:D\to E}\Pi_{a:A}\Pi_{t:B(a)\to D}\big(f(d(a,t)) = e(a, f\circ t)\big)$$

Definition 50. *For algebras* $\mathcal{X} : \mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_j}$ *and* $\mathcal{Y} : \mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_k}$*, define a type family over the type* $\mathsf{W}_0\text{-}\mathsf{Hom}\ \pi_1(\mathcal{X})\ \pi_1(\mathcal{Y})$ *by*

$$\mathsf{W}_1\text{-}\mathsf{Hom}\ (D,d,p)\ (E,e,q)\ (f,\beta) :\equiv \Pi_{c:C}\Pi_{t:B(\mathtt{l}\ c)\to D}\Pi_{s:B(\mathtt{r}\ c)\to D}$$
$$\left(\mathsf{ap}_f(p(c,t,s)) = \beta(\mathtt{l}\ c,t)\cdot q(c,f\circ t,f\circ s)\cdot \beta(\mathtt{r}\ c,s)^{-1}\right)$$

Definition 51. *For algebras* $\mathcal{X} : \mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_j}$ *and* $\mathcal{Y} : \mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_k}$*, define the type of* W-suspension homomorphisms *from* $\mathcal{X}$ *to* $\mathcal{Y}$ *by*

$$\mathsf{W}\text{-}\mathsf{Hom}\ \mathcal{X}\ \mathcal{Y} :\equiv \left(\Sigma\mu_0 : \mathsf{W}_0\text{-}\mathsf{Hom}\ \pi_1(\mathcal{X})\ \pi_1(\mathcal{Y})\right)\ \mathsf{W}_1\text{-}\mathsf{Hom}\ \mathcal{X}\ \mathcal{Y}\ \mu_0$$

Pictorially, the last component of a W-suspension homomorphism witnesses the following commuting diagram for any c, t, s:

$$\begin{array}{ccc}
f(d(\mathtt{l}\ c,t)) & \xrightarrow{\ \mathsf{ap}_f(p(c,t,s))\ } & f(d(\mathtt{r}\ c,s)) \\
\beta(\mathtt{l}\ c,t)\Big| & & \Big|\beta(\mathtt{r}\ c,s) \\
e(\mathtt{l}\ c,f\circ t) & \xrightarrow[\ q(c,f\circ t,f\circ s)\ ]{} & e(\mathtt{r}\ c,f\circ s)
\end{array}$$

Definition 52. *For algebras* $\mathcal{X}_0 : \mathsf{W}_0\text{-}\mathsf{Alg}_{\mathcal{U}_j}$ *and* $\mathcal{Y}_0 : \mathsf{W}_0\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_k}\ \mathcal{X}_0$*, define the type of* fibered W-tree homomorphisms *from* $\mathcal{X}_0$ *to* $\mathcal{Y}_0$ *by*

$$\mathsf{W}_0\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\ (D,d)\ (E,e) :\equiv \Sigma_{f:(\Pi x:D)E(x)}\Pi_{a:A}\Pi_{t:B(a)\to D}\big(f(d(a,t)) = e(a,t,f\circ t)\big)$$

Definition 53. *For algebras* $\mathcal{X}$: W-Alg$_{\mathcal{U}_j}$ *and* $\mathcal{Y}$: W-Fib-Alg$_{\mathcal{U}_k}$ $\mathcal{X}$*, define a type family over the type* W$_0$-Fib-Hom $\pi_1(\mathcal{X})$ $\pi_1(\mathcal{Y})$ *by*

$$\mathsf{W_1\text{-}Fib\text{-}Hom}\ (D,d,p)\ (E,e,q)\ (f,\beta) :\equiv \Pi_{c:C}\Pi_{t:B(\mathtt{l}\ c)\to D}\Pi_{s:B(\mathtt{r}\ c)\to D}$$
$$\Big(\mathsf{dap}_f(p(c,t,s)) = \mathsf{ap}_{p(c,t,s)^E_*}(\beta(\mathtt{l}\ c,t)) \cdot q(c,t,s,f\circ t,f\circ s) \cdot \beta(\mathtt{r}\ c,s)^{-1}\Big)$$

Definition 54. *For algebras* $\mathcal{X}$: W-Alg$_{\mathcal{U}_j}$ *and* $\mathcal{Y}$: W-Fib-Alg$_{\mathcal{U}_k}$ $\mathcal{X}$*, define the type of* fibered W-suspension homomorphisms *from* $\mathcal{X}$ *to* $\mathcal{Y}$ *by*

$$\mathsf{W\text{-}Fib\text{-}Hom}\ \mathcal{X}\ \mathcal{Y} :\equiv \Big(\Sigma\mu_0 : \mathsf{W_0\text{-}Fib\text{-}Hom}\ \pi_1(\mathcal{X})\ \pi_1(\mathcal{Y})\Big)\ \mathsf{W_1\text{-}Fib\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}\ \mu_0$$

Pictorially, the last component of a W-suspension homomorphism witnesses the following commuting diagram for any c, t, s:

$$\begin{array}{ccc}
p(c,t,s)^E_*(f(d(\mathtt{l}\ c,t))) & \xrightarrow{\ \mathsf{dap}_f(p(c,t,s))\ } & f(d(\mathtt{r}\ c,s)) \\
\Big|\ \mathsf{ap}_{p(c,t,s)^E_*}(\beta(\mathtt{l}\ c,t)) & & \Big|\ \beta(\mathtt{r}\ c,s) \\
p(c,t,s)^E_*(e(\mathtt{l}\ c,t,f\circ t)) & \xrightarrow[\ q(c,t,s,f\circ t,f\circ s)\ ]{} & e(\mathtt{r}\ c,s,f\circ s)
\end{array}$$

The recursion and induction principles for W-suspensions can now be defined as usual:

Definition 55. *An algebra* $\mathcal{X}$: W-Alg$_{\mathcal{U}_j}$ satisfies the recursion principle *on a universe* $\mathcal{U}_k$ *if for any algebra* $\mathcal{Y}$: W-Alg$_{\mathcal{U}_k}$ *there exists a* W*-suspension homomorphism from* $\mathcal{X}$ *to* $\mathcal{Y}$*:*

$$\mathsf{has\text{-}W\text{-}rec}_{\mathcal{U}_k}(\mathcal{X}) :\equiv \big(\Pi\mathcal{Y} : \mathsf{W\text{-}Alg}_{\mathcal{U}_k}\big)\ \mathsf{W\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}$$

Definition 56. *An algebra* $\mathcal{X}$: W-Alg$_{\mathcal{U}_j}$ satisfies the induction principle *on a universe* $\mathcal{U}_k$ *if for any fibered algebra* $\mathcal{Y}$: W-Alg$_{\mathcal{U}_k}$ $\mathcal{X}$ *there exists a fibered* W*-suspension homomorphism from* $\mathcal{X}$ *to* $\mathcal{Y}$*:*

$$\mathsf{has\text{-}W\text{-}ind}_{\mathcal{U}_k}(\mathcal{X}) :\equiv \big(\Pi\mathcal{Y} : \mathsf{W\text{-}Fib\text{-}Alg}_{\mathcal{U}_k}\ \mathcal{X}\big)\ \mathsf{W\text{-}Fib\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}$$

We will also need the following uniqueness properties which state that any two (fibered) homomorphisms into any (fibered) algebra $\mathcal{Y}$ are equal:

Definition 57. *An algebra* $\mathcal{X}$: W-Alg$_{\mathcal{U}_j}$ satisfies the recursion uniqueness principle *on a universe* $\mathcal{U}_k$ *if for any other algebra* $\mathcal{Y}$: W-Alg$_{\mathcal{U}_k}$ *the type of* W*-suspension homomorphisms from* $\mathcal{X}$ *to* $\mathcal{Y}$ *is a mere proposition:*

$$\mathsf{has\text{-}W\text{-}rec\text{-}uniq}_{\mathcal{U}_k}(\mathcal{X}) :\equiv \big(\Pi\mathcal{Y} : \mathsf{W\text{-}Alg}_{\mathcal{U}_k}\big)\ \mathsf{is\text{-}prop}(\mathsf{W\text{-}Hom}\ \mathcal{X}\ \mathcal{Y})$$

Definition 58. *An algebra* $\mathcal{X}$: W-Alg$_{\mathcal{U}_j}$ satisfies the induction uniqueness principle *on a universe* $\mathcal{U}_k$ *if for any fibered algebra* $\mathcal{Y}$: W-Alg$_{\mathcal{U}_k}$ $\mathcal{X}$ *the type of* W*-suspension homomorphisms from* $\mathcal{X}$ *to* $\mathcal{Y}$ *is a mere proposition:*

$$\mathsf{has\text{-}W\text{-}ind\text{-}uniq}_{\mathcal{U}_k}(\mathcal{X}) :\equiv \big(\Pi\mathcal{Y} : \mathsf{W\text{-}Fib\text{-}Alg}_{\mathcal{U}_k}\ \mathcal{X}\big)\ \mathsf{is\text{-}prop}(\mathsf{W\text{-}Fib\text{-}Hom}\ \mathcal{X}\ \mathcal{Y})$$

We now define the key concept of homotopy-initiality [2], which translates the notion of existence plus uniqueness into the homotopy type-theoretic setting as contractibility:

Definition 59. *An algebra $\mathcal{X}$: W-$\mathsf{Alg}_{\mathcal{U}_j}$ is* homotopy-initial *on a universe $\mathcal{U}_k$ if for any other algebra $\mathcal{Y}$: W-$\mathsf{Alg}_{\mathcal{U}_k}$ the type of* W*-suspension homomorphisms from $\mathcal{X}$ to $\mathcal{Y}$ is contractible:*

$$\text{is-W-hinit}_{\mathcal{U}_k}(\mathcal{X}) :\equiv \big(\Pi \mathcal{Y} : \text{W-Alg}_{\mathcal{U}_k}\big)\ \text{is-contr}(\text{W-Hom}\ \mathcal{X}\ \mathcal{Y})$$

Lemma 60. *For any $\mathcal{X}$: W-$\mathsf{Alg}_{\mathcal{U}_j}$ we have*

$$\text{is-W-hinit}_{\mathcal{U}_k}(\mathcal{X}) \simeq \text{has-W-rec}_{\mathcal{U}_k}(\mathcal{X}) \times \text{has-W-rec-uniq}_{\mathcal{U}_k}(\mathcal{X})$$

5.2 Main result

Our main result establishes the equivalence between the universal property of being homotopy-initial and the satisfaction of the induction principle:

Theorem 61. *For $A, C : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, $\mathtt{l}, \mathtt{r} : C \to A$, the following conditions on an algebra $\mathcal{X} : \text{W-Alg}_{\mathcal{U}_j}(A, B, C, \mathtt{l}, \mathtt{r})$ are equivalent:*

- *$\mathcal{X}$ satisfies the induction principle on the universe $\mathcal{U}_k$*
- *$\mathcal{X}$ is homotopy-initial on the universe $\mathcal{U}_k$*

for $k \geq j$. In other words, we have

$$\text{has-W-ind}_{\mathcal{U}_k}(\mathcal{X}) \simeq \text{is-W-hinit}_{\mathcal{U}_k}(\mathcal{X})$$

provided $k \geq j$. Moreover, the two types above are mere propositions.

By Lem. 60, homotopy-intiality is equivalent to the principles of recursion plus recursion uniqueness. The uniqueness condition is necessary since in general, the recursion principle does not fully determine an inductive type: the recursion principle for the circle, for example, is also satisfied by the disjoint union of *two* circles.

Before we proceed to the proof of the general case, we look at the analogue of the main theorem for propositional truncations. We can define homotopy-initial $||A||$-algebras as expected:

Definition 62. *An algebra $\mathcal{X}$: $||A||$-$\mathsf{Alg}_{\mathcal{U}_j}$ is* homotopy-initial *on a universe $\mathcal{U}_k$ if for any other algebra $\mathcal{Y}$: $||A||$-$\mathsf{Alg}_{\mathcal{U}_k}$ the type of $||A||$-homomorphisms from $\mathcal{X}$ to $\mathcal{Y}$ is contractible:*

$$\text{is-}||A||\text{-hinit}_{\mathcal{U}_k}(\mathcal{X}) :\equiv \big(\Pi \mathcal{Y} : ||A||\text{-Alg}_{\mathcal{U}_k}\big)\ \text{is-contr}(||A||\text{-Hom}\ \mathcal{X}\ \mathcal{Y})$$

Since we operate in the setting of mere propositions, we do not have to formulate an analogous uniqueness condition, which is uniquely satisfied and hence redundant. Instead, we have:

Lemma 63. *For any $\mathcal{X}$: $||A||$-$\mathsf{Alg}_{\mathcal{U}_j}$ we have*

$$\text{is-}||A||\text{-hinit}_{\mathcal{U}_k}(\mathcal{X}) \simeq \text{has-}||A||\text{-rec}_{\mathcal{U}_k}(\mathcal{X})$$

Thus, Lem. 42 is the analogue of our main result for truncations.

Proof outline A crucial step of the proof is the characterization of the path space $\mu = \nu$ between two (fibered) W-suspension homomorphisms $\mu, \nu : \mathcal{X} \to \mathcal{Y}$ in a more explicit form. For simplicity we only consider the non-fibered case here. We recall that a homomorphism between two algebras (D, d, p), (E, e, q) is a triple (f, β, θ), where $f : C \to D$ is a function between the carrier types, β specifies the behavior of f on the 0-cells, i.e., the value of $f(d(a, t))$, and θ specifies the behavior of f on the 1-cells, i.e., the value of $\mathsf{ap}_f(p(c, t, s))$.

Using the characterization of paths between tuples together with function extensionality, the path space $(f, \beta, \theta) = (g, \gamma, \phi)$ between two homomorphisms should be equivalent to a type of triples (α, η, ψ), where $\alpha : f \sim g$ is a homotopy relating the two underlying mappings, and η, ψ relate β to γ resp. θ to ϕ in an appropriate way. We will call such a triple (α, η, ψ) a W-*suspension cell*. The recursion uniqueness condition on an algebra $\mathcal{X}$ can then be equivalently expressed as saying that for any algebra $\mathcal{Y}$ and homomorphisms μ, ν from $\mathcal{X}$ to $\mathcal{Y}$, there exists a W-suspension cell between μ and ν.

We point out that this uniqueness condition can itself be understood as a certain form of induction, albeit a very specific one. The existence of a W-suspension cell between any two homomorphisms (f, β, θ), (g, γ, ϕ) in particular guarantees the existence of a dependent function $\alpha : \Pi_{x:X}(f(x) = g(x))$ - the "inductor". The behavior of α on the 0-cells, i.e., the value of $\alpha(d(a, t))$, is specified by the term η, which thus serves as a witness for the first "computation rule". Finally, the behavior of α on the 1-cells, i.e., the value of $\mathsf{dap}_\alpha(p(c, t, s))$, is specified by the term ψ, which hence serves as a witness for the second "computation rule." We observe the same pattern in the case of propositional truncations: homomorphisms between $||A||$-algebras are just maps between the carrier types, hence there are no "computation rules" to speak of. A cell between $||A||$-homomorphisms f and g would just be a homotopy $\alpha : \Pi_{x:X}(f(x) = g(x))$ - the "inductor". The existence of such α is of course a moot point since we work with mere propositions.

We can now see why the full induction principle for W-suspensions gives homotopy-initiality: the latter essentially amounts to the recursion principle plus a specific form of induction, both of which are implied by the general induction principle. The hardest part of the proof is showing the converse, i.e., that the general induction principle can be recovered from homotopy-initiality.

We are now ready to give the formal definition of a W-suspension cell. There is an analogous definition of a *fibered* W-*suspension cell*, which uses the fibered versions of W-algebras and homomorphisms. As before, we first introduce the corresponding notions for W-trees and then proceed to the general case of W-suspensions.

Definition 64. *Given algebras* $\mathcal{X}_0 : \mathsf{W}_0\text{-}\mathsf{Alg}_{\mathcal{U}_j}$, $\mathcal{Y}_0 : \mathsf{W}_0\text{-}\mathsf{Alg}_{\mathcal{U}_k}$, *and homomorphisms* $\mu_0, \nu_0 : \mathsf{W}_0\text{-}\mathsf{Hom}\ \mathcal{X}_0\ \mathcal{Y}_0$, *define the type of* W-tree cells *between* μ_0 *and* ν_0 *by*

$$\begin{aligned}&\mathsf{W}_0\text{-}\mathsf{Cell}\ (D, d)\ (E, e)\ (f, \beta)\ (g, \gamma) :\equiv \\ &\qquad \mathsf{W}_0\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\ (D, d)\ \Big(x \mapsto f(x) = g(x); a, t, u \mapsto \beta(a, t) \cdot \mathsf{ap}_{e(a)}({}^{\Pi}\mathsf{E}^{=}(u)) \cdot \gamma(a, t)^{-1}\Big)\end{aligned}$$

Pictorially, the second component of a W-tree cell witnesses the following commuting diagram for any a, t:

$$\begin{array}{ccc}
f(d(a,t)) & \overset{\alpha(d(a,t))}{\rule{3cm}{0.4pt}} & g(d(a,t)) \\
\beta(a,t)\Big| & & \Big|\gamma(a,t) \\
e(a, f\circ t) & \underset{\mathsf{ap}_{e(a)}({}^{\Pi}\mathsf{E}^{=}(\alpha\circ t))}{\rule{3cm}{0.4pt}} & e(a, g\circ t)
\end{array}$$

Definition 65. *Given algebras* $\mathcal{X}_0 : \mathsf{W}_0\text{-}\mathsf{Alg}_{\mathcal{U}_j}$, $\mathcal{Y}_0 : \mathsf{W}_0\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_k}\ \mathcal{X}_0$, *and fibered homomorphisms* $\mu_0, \nu_0 : \mathsf{W}_0\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\ \mathcal{X}_0\ \mathcal{Y}_0$, *define the type of* fibered W-tree cells *between* μ_0 *and* ν_0 *by*

$$\begin{aligned}
&\mathsf{W}_0\text{-}\mathsf{Fib}\text{-}\mathsf{Cell}\ (D,d)\ (E,e)\ (f,\beta)\ (g,\gamma) :\equiv \\
&\qquad \mathsf{W}_0\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\ (D,d)\ \Big(x \mapsto f(x) = g(x); a,t,u \mapsto \beta(a,t) \cdot \mathsf{ap}_{e(a,t)}({}^{\Pi}\mathsf{E}^{=}(u)) \cdot \gamma(a,t)^{-1}\Big)
\end{aligned}$$

Pictorially, the second component of a W-tree cell witnesses the following commuting diagram for any a, t:

$$\begin{array}{ccc}
f(d(a,t)) & \overset{\alpha(d(a,t))}{\rule{3cm}{0.4pt}} & g(d(a,t)) \\
\beta(a,t)\Big| & & \Big|\gamma(a,t) \\
e(a, t, f\circ t) & \underset{\mathsf{ap}_{e(a,t)}({}^{\Pi}\mathsf{E}^{=}(\alpha\circ t))}{\rule{3cm}{0.4pt}} & e(a, t, g\circ t)
\end{array}$$

We will generally omit all but the last two arguments to W_0-Cell and W_0-Fib-Cell.

Following the same methodology, we postulate that a W-suspension cell between (f, β, θ) and (g, γ, ϕ) should consist of a W-tree cell (α, η) together with a proof that the value of $\mathsf{dap}_\alpha(p(c,t,s))$ is the "obvious" one. However, the type of the term $\mathsf{dap}_\alpha(p(c,t,s))$ involves a transport along the fibers of the type family $x \mapsto f(x) = g(x)$, making it unwieldy to work with. Instead, we axiomatize the value of $\mathsf{nat}(\alpha, p(c,t,s))$, which nevertheless specifies the value of $\mathsf{dap}_\alpha(p(c,t,s))$ uniquely as the latter term is expressible using the former.

Determining (and stating!) what the value of $\mathsf{nat}(\alpha, p(c,t,s))$ should be requires a little work; this is expected since we are now working with paths on a higher level. To state the crucial definitions more compactly, we introduce the following notations.

Definition 66. *Given*

- *functions* $e_1 : X_1 \to Y$, $e_2 : X_2 \to Y$
- *a heterogeneous homotopy* $q : e_1 \sim_{\mathcal{H}} e_2$
- *paths* $r_1 : a_1 =_{X_1} b_1$, $r_2 : a_2 =_{X_2} b_2$ *and* $\delta_1 : c_1 =_Y c_2$, $\delta_2 : d_1 =_Y d_2$
- *paths* $\beta_1 : c_1 =_Y e_1(a_1)$, $\beta_2 : c_2 =_Y e_2(a_2)$ *and* $\gamma_1 : d_1 =_Y e_1(b_1)$, $\gamma_2 : d_2 =_Y e_2(b_2)$

- *higher paths* $\Theta : \delta_1 = \beta_1 \cdot q(a_1, a_2) \cdot \beta_2^{-1}$ *and* $\Phi : \delta_2 = \gamma_1 \cdot q(b_1, b_2) \cdot \gamma_2^{-1}$

we let $\mathcal{P}(q, r_1, r_2, \Theta, \Phi)$ *be the higher path in Fig. 1.*

Definition 67. *Given*

- *a function* $F : Y_1 \to Y_2$
- *functions* $e_1 : X_1 \to Y_1$, $e_2 : X_2 \to Y_2$
- *a heterogeneous homotopy* $q : (F \circ e_1) \sim_{\mathcal{H}} e_2$
- *paths* $r_1 : a_1 =_{X_1} b_1$, $r_2 : a_2 =_{X_2} b_2$ *and* $\delta_1 : F(c_1) =_{Y_2} c_2$, $\delta_2 : F(d_1) =_{Y_2} d_2$
- *paths* $\beta_1 : c_1 =_{Y_1} e_1(a_1)$, $\beta_2 : c_2 =_{Y_2} e_2(a_2)$ *and* $\gamma_1 : d_1 =_{Y_1} e_1(b_1)$, $\gamma_2 : d_2 =_{Y_2} e_2(b_2)$
- *higher paths* $\Theta : \delta_1 = \mathsf{ap}_F(\beta_1) \cdot q(a_1, a_2) \cdot \beta_2^{-1}$ *and* $\Phi : \delta_2 = \mathsf{ap}_F(\gamma_1) \cdot q(b_1, b_2) \cdot \gamma_2^{-1}$

we let $\mathcal{Q}(q, r_1, r_2, \Theta, \Phi)$ *be the higher path in Fig. 2.*

Definition 68. *For algebras* $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}$, $\mathcal{Y} : \mathsf{W\text{-}Alg}_{\mathcal{U}_k}$ *and homomorphisms* $\mu, \nu : \mathsf{W\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}$, *define a type family over the type* $\mathsf{W_0\text{-}Cell}\ \pi_1(\mu)\ \pi_1(\nu)$ *by*

$$\mathsf{W_1\text{-}Cell}\ (D, d, p)\ (E, e, q)\ (f, \beta, \theta)\ (g, \gamma, \phi)\ (\alpha, \eta) :\equiv$$
$$\Pi_{c:C}\Pi_{t:B(\mathtt{l}\ c)\to D}\Pi_{s:B(\mathtt{r}\ c)\to D}\Bigg(\mathsf{nat}(\alpha, p_{c,t,s}) = \mathsf{ap}_{-\cdot\mathsf{ap}_g(p(c,t,s))}(\eta(\mathtt{l}\ c, t)) \cdot$$
$$\mathcal{P}\Big(q_c, {}^{\Pi}\mathsf{E}^{=}(\alpha \circ t), {}^{\Pi}\mathsf{E}^{=}(\alpha \circ s), \theta_{c,t,s}, \phi_{c,t,s}\Big) \cdot \Big(\mathsf{ap}_{\mathsf{ap}_f(p(c,t,s))\cdot -}(\eta(\mathtt{r}\ c, s))\Big)^{-1}\Bigg)$$

Definition 69. *For algebras* $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}$, $\mathcal{Y} : \mathsf{W\text{-}Alg}_{\mathcal{U}_k}\ \mathcal{X}$ *and fibered homomorphisms* $\mu, \nu :$ $\mathsf{W\text{-}Fib\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}$, *define a type family over the type* $\mathsf{W_0\text{-}Fib\text{-}Cell}\ \pi_1(\mu)\ \pi_1(\nu)$ *by*

$$\mathsf{W_1\text{-}Fib\text{-}Cell}\ (D, d, p)\ (E, e, q)\ (f, \beta, \theta)\ (g, \gamma, \phi)\ (\alpha, \eta) :\equiv$$
$$\Pi_{c:C}\Pi_{t:B(\mathtt{l}\ c)\to D}\Pi_{s:B(\mathtt{r}\ c)\to D}\Bigg(\mathsf{nat}_{\mathcal{F}}(\alpha, p_{c,t,s}) = \mathsf{ap}_{\mathsf{ap}_{p(c,t,s)^E_*}(-)\cdot\mathsf{dap}_g(p(c,t,s))}(\eta(\mathtt{l}\ c, t)) \cdot$$
$$\mathcal{P}\Big(q_{c,t,s}, {}^{\Pi}\mathsf{E}^{=}(\alpha \circ t), {}^{\Pi}\mathsf{E}^{=}(\alpha \circ s), \theta_{c,t,s}, \phi_{c,t,s}\Big) \cdot \Big(\mathsf{ap}_{\mathsf{dap}_f(p(c,t,s))\cdot -}(\eta(\mathtt{r}\ c, s))\Big)^{-1}\Bigg)$$

We will usually leave out all but the last three arguments to $\mathsf{W_1\text{-}Cell}$ and $\mathsf{W_1\text{-}Fib\text{-}Cell}$.

Definition 70. *For algebras* $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}$, $\mathcal{Y} : \mathsf{W\text{-}Alg}_{\mathcal{U}_k}$ *and homomorphisms* $\mu, \nu : \mathsf{W\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}$, *define the type of* W-suspension cells *between* μ *and* ν *by*

$$\mathsf{W\text{-}Cell}\ \mu\ \nu :\equiv \Big(\Sigma\mathcal{C}_0 : \mathsf{W_0\text{-}Cell}\ \pi_1(\mu)\ \pi_1(\nu)\Big)\ \mathsf{W_1\text{-}Cell}\ \mu\ \nu\ \mathcal{C}_0$$

Definition 71. *For algebras* $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}$, $\mathcal{Y} : \mathsf{W\text{-}Alg}_{\mathcal{U}_k}\ \mathcal{X}$ *and fibered homomorphisms* $\mu, \nu : \mathsf{W\text{-}Fib\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}$*, define the type of* fibered W-suspension cells *between* μ *and* ν *by*

$$\mathsf{W\text{-}Fib\text{-}Cell}\ \mu\ \nu :\equiv \Big(\Sigma \mathcal{C}_0 : \mathsf{W_0\text{-}Fib\text{-}Cell}\ \pi_1(\mu)\ \pi_1(\nu)\Big)\ \mathsf{W_1\text{-}Fib\text{-}Cell}\ \mu\ \nu\ \mathcal{C}_0$$

Pictorially, the last component of a W-suspension cell witnesses the commuting diagram in Fig. 3 for any c, t, s. The last component of a fibered W-suspension cell witnesses the commuting diagram in Fig. 4 for any c, t, s.

Lemma 72. *For algebras* $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}$, $\mathcal{Y} : \mathsf{W\text{-}Alg}_{\mathcal{U}_k}$ *and homomorphisms* $\mu, \nu : \mathsf{W\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}$*, the path space* $\mu = \nu$ *is equivalent to the type of* W*-suspension cells between* μ *and* ν*:*

$$\mu = \nu\ \simeq\ \mathsf{W\text{-}Cell}\ \mu\ \nu$$

Lemma 73. *For algebras* $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}$, $\mathcal{Y} : \mathsf{W\text{-}Alg}_{\mathcal{U}_k}\ \mathcal{X}$ *and fibered homomorphisms* $\mu, \nu : \mathsf{W\text{-}Fib\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}$*, the path space* $\mu = \nu$ *is equivalent to the type of fibered* W*-suspension cells between* μ *and* ν*:*

$$\mu = \nu\ \simeq\ \mathsf{W\text{-}Fib\text{-}Cell}\ \mu\ \nu$$

For the proof see Sect. A. The proof of the main result now consists of the following steps:

1) *Show that the induction principle implies the recursion principle, that is*:

$$\mathsf{has\text{-}W\text{-}ind}_{\mathcal{U}_k}(\mathcal{X}) \to \mathsf{has\text{-}W\text{-}rec}_{\mathcal{U}_k}(\mathcal{X})$$

for any $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}$. See Sect. B.

2) *Show that the induction principle implies both uniqueness conditions, that is*:

$$\mathsf{has\text{-}W\text{-}ind}_{\mathcal{U}_k}(\mathcal{X}) \to \mathsf{has\text{-}W\text{-}ind\text{-}uniq}_{\mathcal{U}_k}(\mathcal{X})$$
$$\mathsf{has\text{-}W\text{-}ind}_{\mathcal{U}_k}(\mathcal{X}) \to \mathsf{has\text{-}W\text{-}rec\text{-}uniq}_{\mathcal{U}_k}(\mathcal{X})$$

for any $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}$. See Sect. C.

3) *Show that the recursion plus recursion uniqueness principles imply the induction principle, that is*:

$$\mathsf{has\text{-}W\text{-}rec}_{\mathcal{U}_k}(\mathcal{X}) \times \mathsf{has\text{-}W\text{-}rec\text{-}uniq}_{\mathcal{U}_k}(\mathcal{X}) \to \mathsf{has\text{-}W\text{-}ind}_{\mathcal{U}_k}(\mathcal{X})$$

for any $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}$. See Sect. D.

Using Lem. 60 we thus have a logical equivalence between $\mathsf{has\text{-}W\text{-}ind}_{\mathcal{U}_k}(\mathcal{X})$ and $\mathsf{is\text{-}W\text{-}hinit}_{\mathcal{U}_k}(\mathcal{X})$. It remains to show that both of these types are mere propositions. The latter is a mere proposition by Lem. 6. To show that $\mathsf{has\text{-}W\text{-}ind}_{\mathcal{U}_k}(\mathcal{X})$ is a mere proposition, it is sufficient to do so under the assumption that it is inhabited. Since $\mathcal{X}$ satisfies the induction principle, by the second step it satisfies the induction uniqueness principle. This means that for any fibered algebra $\mathcal{Y}$, the type $\mathsf{W\text{-}Fib\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}$ is a mere proposition. Since a family of mere propositions is itself a mere proposition, this finishes the proof.

$$
\begin{array}{c}
\left(\beta_1 \cdot \mathsf{ap}_{e_1}(r_1) \cdot \gamma_1^{-1}\right) \cdot \delta_2 \\
\Big| \\
\left(\beta_1 \cdot \mathsf{ap}_{e_1}(r_1)\right) \cdot \left(\gamma_1^{-1} \cdot \delta_2\right) \\
\Big| \; \textit{via } \mathbf{I}^2_{\square}(\Phi) \\
\left(\beta_1 \cdot \mathsf{ap}_{e_1}(r_1)\right) \cdot \left(q(b_1, b_2) \cdot \gamma_2^{-1}\right) \\
\Big| \\
\beta_1 \cdot \left(\mathsf{ap}_{e_1}(r_1) \cdot q(b_1, b_2)\right) \cdot \gamma_2^{-1} \\
\Big| \; \textit{via } \mathsf{nat}_{\mathcal{H}}(q, r_1, r_2)^{-1} \\
\beta_1 \cdot \left(q(a_1, a_2) \cdot \mathsf{ap}_{e_2}(r_2)\right) \cdot \gamma_2^{-1} \\
\Big| \\
\left(\beta_1 \cdot q(a_1, a_2)\right) \cdot \left(\mathsf{ap}_{e_2}(r_2) \cdot \gamma_2^{-1}\right) \\
\Big| \; \textit{via } \mathbf{I}^1_{\square}(\Theta)^{-1} \\
\left(\delta_1 \cdot \beta_2\right) \cdot \left(\mathsf{ap}_{e_1}(r_2) \cdot \gamma_2^{-1}\right) \\
\Big| \\
\delta_1 \cdot \left(\beta_2 \cdot \mathsf{ap}_{e_2}(r_2) \cdot \gamma_2^{-1}\right)
\end{array}
$$

Figure 1: The path $\mathcal{P}(q, r_1, r_2, \Theta, \Phi)$

$$
\begin{array}{c}
\mathsf{ap}_F\left(\beta_1 \cdot \mathsf{ap}_{e_1}(r_1) \cdot \gamma_1^{-1}\right) \cdot \delta_2 \\
\Big| \\
\Big(\mathsf{ap}_F(\beta_1) \cdot \mathsf{ap}_F(\mathsf{ap}_{e_1}(r_1)) \cdot (\mathsf{ap}_F(\gamma_1))^{-1}\Big) \cdot \delta_2 \\
\Big| \\
\Big(\mathsf{ap}_F(\beta_1) \cdot \mathsf{ap}_{F \circ e_1}(r_1) \cdot (\mathsf{ap}_F(\gamma_1))^{-1}\Big) \cdot \delta_2 \\
\Big| \; \mathcal{P}(q, r_1, r_2, \Theta, \Phi) \\
\delta_1 \cdot \left(\beta_2 \cdot \mathsf{ap}_{e_2}(r_2) \cdot \gamma_2^{-1}\right)
\end{array}
$$

Figure 2: The path $\mathcal{Q}(q, r_1, r_1, \Theta, \Phi)$

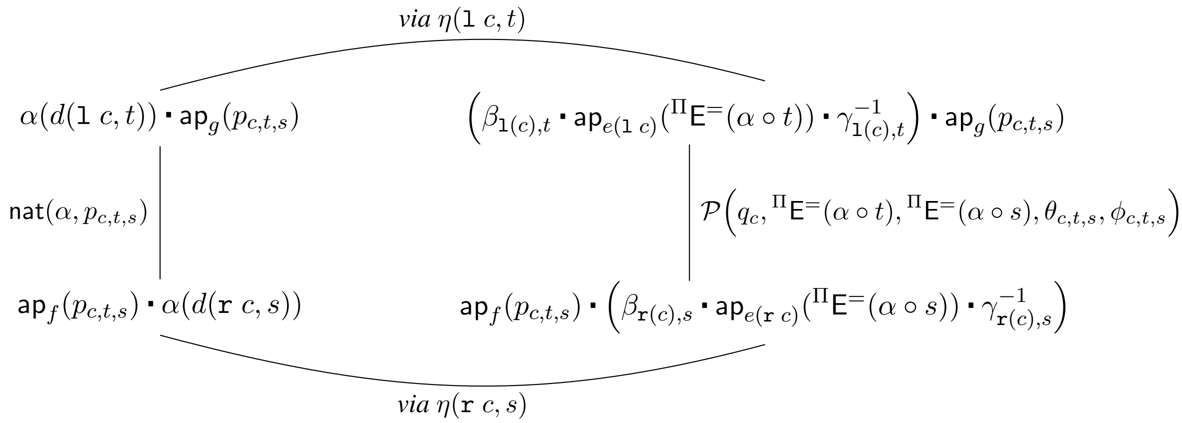


Figure 3: Commuting diagram for Def. 70

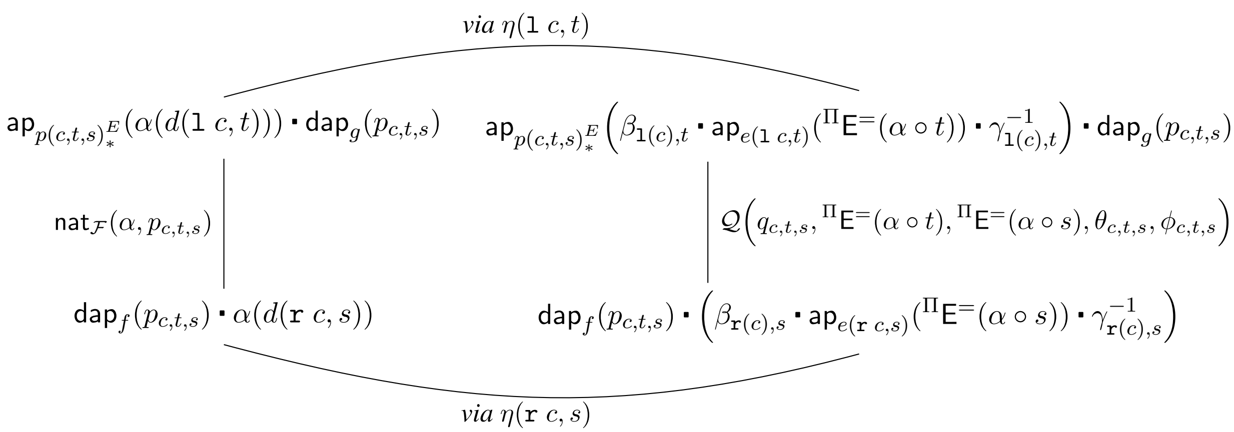


Figure 4: Commuting diagram for Def. 71

5.3 Definability

We now show how to derive the analogue of our main result for the circle $\mathbb{S}$; the cases of $\mathbb{S}_a$ and other inductive types presentable as W-suspensions follow the same methodology. In the rest of this section we work with a specific W-suspension $\mathsf{W}(A, B, C, \mathtt{l}, \mathtt{r}) : \mathcal{U}_0$ where $A, C :\equiv \mathbf{1}$, $B :\equiv \lambda_{\_:\mathbf{1}}\mathbf{0}, \mathtt{l}, \mathtt{r} :\equiv \lambda_{\_:\mathbf{1}}\star$.

Definition 74. *An algebra $\mathcal{X} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_j}$ is* homotopy-initial *on a universe $\mathcal{U}_k$ if for any other algebra $\mathcal{Y} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_k}$ the type of $\mathbb{S}$-homomorphisms from $\mathcal{X}$ to $\mathcal{Y}$ is contractible:*

$$\mathsf{is}\text{-}\mathbb{S}\text{-}\mathsf{hinit}_{\mathcal{U}_k}(\mathcal{X}) :\equiv \big(\Pi \mathcal{Y} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_k}\big)\ \mathsf{is}\text{-}\mathsf{contr}(\mathbb{S}\text{-}\mathsf{Hom}\ \mathcal{X}\ \mathcal{Y})$$

Lemma 75. *We have a function*

$$\mathbb{S}\text{-}\mathsf{to}\text{-}\mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_i} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_i} \to \mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$$

which is an equivalence.

Proof. We define the function by the mapping

$$(D, d, p) \mapsto \big(D, \lambda_a\lambda_t d, \lambda_c\lambda_t\lambda_s p\big)$$

It is not hard to see that this is an equivalence. □

Lemma 76. *For any algebra $\mathcal{X} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ we have a function*

$$\mathbb{S}\text{-}\mathsf{to}\text{-}\mathsf{W}\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_i}(\mathcal{X}) : \mathbb{S}\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_i}\ \mathcal{X} \to \mathsf{W}\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_i}\ \big(\mathbb{S}\text{-}\mathsf{to}\text{-}\mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_i}\ \mathcal{X}\big)$$

which is an equivalence.

Proof. Fix an algebra $(D, d, p) : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$. We define the function by the mapping

$$(E, e, q) \mapsto \big(E, \lambda_a\lambda_t\lambda_u e, \lambda_c\lambda_t\lambda_u\lambda_s\lambda_v q\big)$$

It is not hard to see that this is an equivalence. □

Lemma 77. *For any algebras $\mathcal{X} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$, $\mathcal{Y} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_j}$ we have*

$$\mathbb{S}\text{-}\mathsf{Hom}\ \mathcal{X}\ \mathcal{Y} \simeq \mathsf{W}\text{-}\mathsf{Hom}\ \big(\mathbb{S}\text{-}\mathsf{to}\text{-}\mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_i}\ \mathcal{X}\big)\ \big(\mathbb{S}\text{-}\mathsf{to}\text{-}\mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_i}\ \mathcal{Y}\big)$$

Lemma 78. *For any algebras $\mathcal{X} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$, $\mathcal{Y} : \mathbb{S}\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_j}\ \mathcal{X}$ we have*

$$\mathbb{S}\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\ \mathcal{X}\ \mathcal{Y} \simeq \mathsf{W}\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\ \big(\mathbb{S}\text{-}\mathsf{to}\text{-}\mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_i}\ \mathcal{X}\big)\ \big(\mathbb{S}\text{-}\mathsf{to}\text{-}\mathsf{W}\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_i}(\mathcal{X})\ \mathcal{Y}\big)$$

Lemma 79. *For any $\mathcal{X} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ we have*

$$\mathsf{has}\text{-}\mathbb{S}\text{-}\mathsf{rec}_{\mathcal{U}_j}(\mathcal{X}) \simeq \mathsf{has}\text{-}\mathsf{W}\text{-}\mathsf{rec}_{\mathcal{U}_j}(\mathbb{S}\text{-}\mathsf{to}\text{-}\mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_i}(\mathcal{X}))$$
$$\mathsf{has}\text{-}\mathbb{S}\text{-}\mathsf{ind}_{\mathcal{U}_j}(\mathcal{X}) \simeq \mathsf{has}\text{-}\mathsf{W}\text{-}\mathsf{ind}_{\mathcal{U}_j}(\mathbb{S}\text{-}\mathsf{to}\text{-}\mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_i}(\mathcal{X}))$$
$$\mathsf{is}\text{-}\mathbb{S}\text{-}\mathsf{hinit}_{\mathcal{U}_j}(\mathcal{X}) \simeq \mathsf{is}\text{-}\mathsf{W}\text{-}\mathsf{hinit}_{\mathcal{U}_j}(\mathbb{S}\text{-}\mathsf{to}\text{-}\mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_i}(\mathcal{X}))$$

Corollary 80. *For an algebra $\mathcal{X} : \mathbb{S}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$, the following conditions are equivalent:*

- *$\mathcal{X}$ satisfies the induction principle on the universe $\mathcal{U}_j$*
- *$\mathcal{X}$ is homotopy-initial on the universe $\mathcal{U}_j$*

In other words, we have

$$\mathsf{has}\text{-}\mathbb{S}\text{-}\mathsf{ind}_{\mathcal{U}_j}(\mathcal{X}) \simeq \mathsf{is}\text{-}\mathbb{S}\text{-}\mathsf{hinit}_{\mathcal{U}_j}(\mathcal{X})$$

Moreover, the two types above are mere propositions.

Proof. We use Lem. 79 and 61. □

6 Conclusion

We have investigated higher inductive types with propositional computational behavior and shown that they can be equivalently characterized as homotopy-initial algebras. We have stated and proved this result for propositional truncations and for the so-called W-suspensions, which subsume a number of other interesting cases - ordinary W-types, the unit circle $\mathbf{S}^1$, the interval type $\mathbf{I}$, all the higher spheres $\mathbf{S}^n$, and all suspensions. The characterization of these individual types as homotopy-initial algebras can be easily obtained as a corollary to our main theorem. Furthermore, we can readily apply the method presented here to obtain an analogous result for set truncations and set quotients. We conjecture that similar results can be established for other higher inductive types - such as homotopy (co)limits, tori, group quotients, or real numbers - following the same methodology. We are planning to formalize the results presented here in the Coq proof assistant.

Finally, we remark that the entire field of Homotopy Type Theory is a subject of intense research and many questions pertaining to higher inductive types and univalence are yet to be satisfactorily answered. Two important open problems are finding a unifying schema for general higher inductive types (see [15] for progress towards this goal) and developing a computational interpretation of HoTT (partially answered by the cubical set model [4]).

Acknowledgment

The author would like to thank her advisors, Profs. Steve Awodey and Frank Pfenning, for their help.

References

[1] S. Awodey and A. Bauer. Propositions as [types]. *Journal of Logic and Computation*, 14(4):447–471, 2004.

[2] S. Awodey, N. Gambino, and K. Sojakova. Inductive types in Homotopy Type Theory. In *Logic in Computer Science (LICS 2012)*, pages 95–104. IEEE Computer Society, 2012.

[3] S. Awodey and M. Warren. Homotopy-theoretic models of identity types. *Mathematical Proceedings of the Cambridge Philosophical Society*, 146(1):45–55, 2009.

[4] M. Bezem, T. Coquand, and S. Huber. A model of type theory in cubical sets, 2014. Available at `http://www.cse.chalmers.se/~coquand/mod1.pdf`.

[5] N. Gambino and R. Garner. The identity type weak factorisation system. *Theoretical Computer Science*, 409(3):94–109, 2008.

[6] R. Garner. Two-dimensional models of type theory. *Mathematical Structures in Computer Science*, 19(4):687–736, 2009.

[7] M. Hofmann and T. Streicher. The groupoid interpretation of type theory. In *Twenty-five years of constructive type theory 1995*, volume 36 of *Oxford Logic Guides*, pages 83–111. Oxford Univ. Press, 1998.

[8] C. Kapulkin, P. Lumsdaine, and V. Voevodsky. The simplicial model of univalent foundations. Available at arxiv.org as arXiv:1211.2851v1, 2012.

[9] N. Kraus. The truncation map $|\_| : \mathbb{N} \to ||\mathbb{N}||$ is nearly invertible, 2013. Post on the Homotopy Type Theory blog.

[10] D. Licata and G. Brunerie. $\pi_n(\mathbf{S}^n)$ in Homotopy Type Theory. In *Certified Programs and Proofs*, volume 8307 of *LNCS*, pages 1–16. Springer, 2013.

[11] D. Licata and R. Harper. Canonicity for 2-dimensional type theory. In *Principles of Programming Languages (POPL 2012)*, pages 337–348. ACM, 2012.

[12] D. Licata and M. Shulman. Calculating the fundamental group of the circle in Homotopy Type Theory. In *Logic in Computer Science (LICS 2013)*, pages 223–232. IEEE Computer Society, 2013.

[13] P. Lumsdaine. Weak ω-categories from intensional type theory. *Logical Methods in Computer Science*, 6:1–19, 2010.

[14] P. Lumsdaine. Higher inductive types: a tour of the menagerie, 2011. Post on the Homotopy Type Theory blog.

[15] P. Lumsdaine and M. Shulman. Semantics of higher inductive types, 2012. Available at `http://uf-ias-2012.wikispaces.com/file/view/semantics.pdf`.

[16] P. Martin-Löf. An intuitionistic theory of types: Predicative part. In *Logic Colloquium 1973*, pages 73–118. North-Holland, 1975.

[17] P. Martin-Löf. Constructive mathematics and computer programming. In *Logic, Methodology, and Philosophy of Science*, pages 153–175. North-Holland, 1982.

[18] U. Norell. *Towards a practical programming language based on dependent type theory*. PhD thesis, Chalmers University of Technology, 2007.

[19] M. Shulman. Homotopy Type Theory, VI, 2011. Post on the n-category cafe blog.

[20] T. Streicher. Investigations into intensional type theory. Habilitation Thesis. Available from the authors web page, 1993.

[21] Coq Development Team. *The Coq Proof Assistant Reference Manual, version 8.4pl3*. INRIA, 2012. Available at `coq.inria.fr`.

[22] The Univalent Foundations Program, Institute for Advanced Study. *Homotopy Type Theory - Univalent Foundations of Mathematics*. Univalent Foundations Project, 2013.

[23] B. van den Berg and R. Garner. Types are weak ω-groupoids. *London Mathematical Society*, 102(2):370–394, 2011.

[24] B. van den Berg and R. Garner. Topological and simplicial models of identity types. *ACM Transactions on Computational Logic*, 13(1), 2012.

[25] V. Voevodsky. Univalent foundations of mathematics, 2011. Invited talk at the Workshop on Logic, Language, Information and Computation (WoLLIC 2011).

[26] M. Warren. *Homotopy-theoretic aspects of constructive type theory*. PhD thesis, Carnegie Mellon University, 2008.

A The path space of homomorphisms

In this section we prove lemmas 72 and 73. We start by showing the analogous statements for the simpler case of W-trees.

Lemma 81. *For algebras $\mathcal{X}_0 : \mathsf{W}_0\text{-}\mathsf{Alg}_{\mathcal{U}_j}$, $\mathcal{Y}_0 : \mathsf{W}_0\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_k}\ \mathcal{X}_0$ and fibered homomorphisms $\mu_0, \nu_0 : \mathsf{W}_0\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\ \mathcal{X}_0\ \mathcal{Y}_0$, there is a function*

$$\mathsf{W}_0\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\text{-}\mathsf{Path}\text{-}\mathsf{to}\text{-}\mathsf{Cell} : \mu_0 = \nu_0 \ \simeq\ \mathsf{W}_0\text{-}\mathsf{Fib}\text{-}\mathsf{Cell}\ \mu_0\ \nu_0$$

which is an equivalence.

Proof. Let the algebras $(D, d) : \mathsf{W}_0\text{-}\mathsf{Alg}_{\mathcal{U}_j}$ and $(E, e) : \mathsf{W}_0\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_k}\ (D, d)$ be given. We define the desired function by path induction: for a homomorphism $(f, \beta) : \mathsf{W}_0\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\ (D, d)\ (E, d)$, we put $\mathsf{refl}(f, \beta) \mapsto (\mathsf{id}_{\mathcal{H}}(f), \eta)$, where $\eta(a, t)$ is the path

$$\begin{array}{c} \mathsf{refl} \\ | \\ \beta(a,t) \cdot \mathsf{refl} \cdot \beta(a,t)^{-1} \\ | \\ \beta(a,t) \cdot \mathsf{ap}_{e(a,t)}({}^{\Pi}\mathsf{E}^{=}(\mathsf{id}_{\mathcal{H}}(f \circ t))) \cdot \beta(a,t)^{-1} \end{array}$$

It is not hard to see that this function is an equivalence. □

Corollary 82. *Given algebras $\mathcal{X}_0 : \mathsf{W}_0\text{-}\mathsf{Alg}_{\mathcal{U}_j}$, $\mathcal{Y}_0 : \mathsf{W}_0\text{-}\mathsf{Alg}_{\mathcal{U}_k}$ and homomorphisms $\mu_0, \nu_0 : \mathsf{W}_0\text{-}\mathsf{Hom}\ \mathcal{X}_0\ \mathcal{Y}_0$, there is a function*

$$\mathsf{W}_0\text{-}\mathsf{Hom}\text{-}\mathsf{Path}\text{-}\mathsf{to}\text{-}\mathsf{Cell} : \mu_0 = \nu_0 \ \simeq\ \mathsf{W}_0\text{-}\mathsf{Cell}\ \mu_0\ \nu_0$$

which is an equivalence.

We now proceed to prove lemmas 72 and 73. Both proofs follow along the same lines; we only treat the dependent case here.

Fix algebras $(D, d, p) : \mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_j}$ and $(E, e, q) : \mathsf{W}\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_k}\ (D, d, p)$ and fibered homomorphisms $(\mu_0, \theta), (\nu_0, \phi) : \mathsf{W}\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\ (D, d, p)\ (E, e, q)$. We establish the following chain of equivalences:

$$\begin{aligned} & (\mu_0, \theta) = (\nu_0, \phi) \\ \simeq\ & \Big(\Sigma \mathcal{C}_0 : \mu_0 = \nu_0\Big)\ (\mathcal{C}_0)_*^{\mathsf{W}_1\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\ (D,d,p)\ (E,e,q)}(\theta) = \phi \\ \simeq\ & \Big(\Sigma \mathcal{C}_0 : \mu_0 = \nu_0\Big)\ \mathsf{W}_1\text{-}\mathsf{Fib}\text{-}\mathsf{Cell}\ (\mu_0, \theta)\ (\nu_0, \phi)\ \big(\mathsf{W}_0\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\text{-}\mathsf{Path}\text{-}\mathsf{to}\text{-}\mathsf{Cell}\ \mathcal{C}_0\big) \\ \simeq\ & \Big(\Sigma \mathcal{C}_0 : \mathsf{W}_0\text{-}\mathsf{Fib}\text{-}\mathsf{Cell}\ \mu_0\ \nu_0\Big)\ \mathsf{W}_1\text{-}\mathsf{Fib}\text{-}\mathsf{Cell}\ (\mu_0, \theta)\ (\nu_0, \phi)\ \mathcal{C}_0 \\ \equiv\ & \mathsf{W}\text{-}\mathsf{Fib}\text{-}\mathsf{Cell}\ (\mu_0, \theta)\ (\nu_0, \phi) \end{aligned}$$

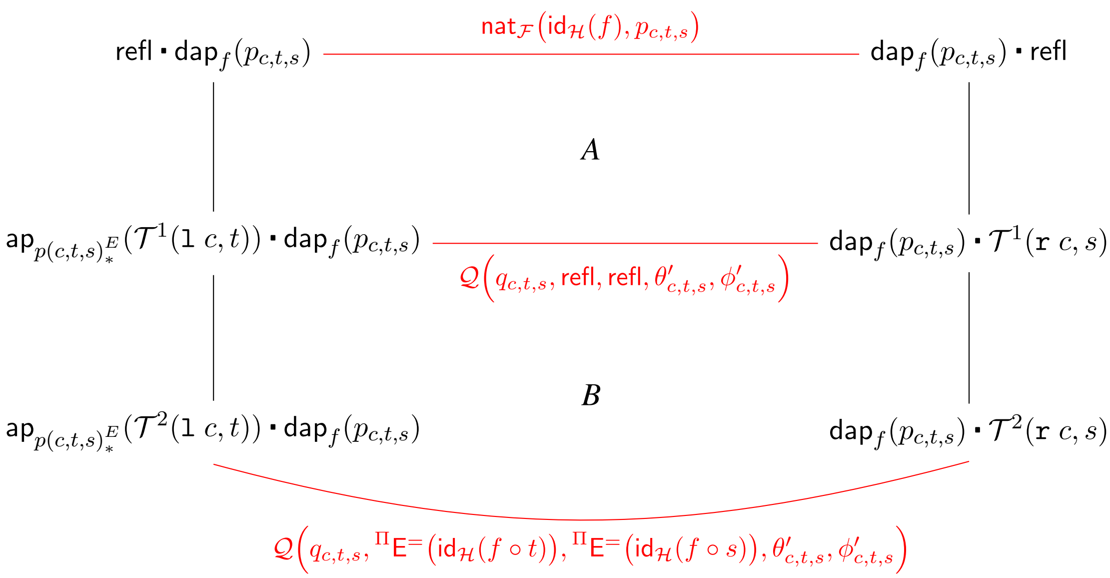


Figure 5: Diagram to be shown equivalent to $\theta'_{c,t,s} = \phi'_{c,t,s}$

The first and third equivalences are clear. To prove the second equivalence, it suffices to establish that for any fibered W-tree homomorphisms μ'_0, ν'_0 : W_0-Fib-Hom (D, d) (E, e), path $\mathcal{C}_0 : \mu'_0 = \nu'_0$, and terms θ' : W_1-Fib-Hom (D, d, p) (E, e, q) μ'_0 and ϕ' : W_1-Fib-Hom (D, d, p) (E, e, q) ν'_0, we have

$$
\begin{aligned}
& (\mathcal{C}_0)_*^{\mathsf{W}_1\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\ (D,d,p)\ (E,e,q)}(\theta') \\
\simeq\ & \mathsf{W}_1\text{-}\mathsf{Fib}\text{-}\mathsf{Cell}\ (\mu'_0, \theta')\ (\nu'_0, \phi')\ \big(\mathsf{W}_0\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\text{-}\mathsf{Path}\text{-}\mathsf{to}\text{-}\mathsf{Cell}\ \mathcal{C}_0\big)
\end{aligned}
$$

We proceed by path induction on $\mathcal{C}_0$. Fix a homomorphism (f, β) : W_0-Fib-Hom (D, d) (E, e) and θ', ϕ' : W_1-Fib-Hom (D, d, p) (E, e, q) (f, β). To show

$$
\theta' = \phi' \quad \simeq \quad \mathsf{W}_1\text{-}\mathsf{Fib}\text{-}\mathsf{Cell}\ (f, \beta, \theta')\ (f, \beta, \phi')\ \big(\mathsf{W}_0\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\text{-}\mathsf{Path}\text{-}\mathsf{to}\text{-}\mathsf{Cell}\ \mathsf{refl}(f, \beta)\big)
$$

it suffices to show that for any c, t, s, the type $\theta'_{c,t,s} = \phi'_{c,t,s}$ is equivalent to the commutativity of the outer rectangle in Fig. 5, where

$$
\begin{aligned}
\mathcal{T}^1(a, t) &:= \beta(a, t) \cdot \mathsf{refl} \cdot \beta(a, t)^{-1} \\
\mathcal{T}^2(a, t) &:= \beta(a, t) \cdot \mathsf{ap}_{e(a,t)}\big({}^{\Pi}\mathsf{E}^{=}\big(\mathsf{id}_{\mathcal{H}}(f \circ t)\big)\big) \cdot \beta(a, t)^{-1}
\end{aligned}
$$

Rectangle B commutes by easy path induction; it therefore suffices to show that $\theta'_{c,t,s} = \phi'_{c,t,s}$ is equivalent to the commutativity of the rectangle A. This is a simple exercise.

B Induction implies recursion

In this section we want to show that

$$\mathsf{has\text{-}W\text{-}ind}_{\mathcal{U}_k}(\mathcal{X}) \to \mathsf{has\text{-}W\text{-}rec}_{\mathcal{U}_k}(\mathcal{X})$$

for any $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}$. Fix an algebra $(D, d, p) : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}$ and assume that $\mathsf{has\text{-}W\text{-}ind}_{\mathcal{U}_k}(D, d, p)$ holds. To show that $\mathsf{has\text{-}W\text{-}rec}_{\mathcal{U}_k}(D, d, p)$ holds, fix any other algebra $(E, e, q) : \mathsf{W\text{-}Alg}_{\mathcal{U}_k}$. In order to apply the induction principle, we need to turn this into a fibered algebra (E', e', q'). The first two components are easy: put $E'(x) :\equiv E$ and $e'(a, t, u) :\equiv e(a, u)$. For the last component, we note that the transport between any two fibers of a constant type family is constant. We can thus define $q'(c, t, s, u, v)$ to be the path

$$(p_{c,t,s})_*^{-\mapsto Y}(e(\mathtt{l}\ c, u)) \;\text{———}\; e(\mathtt{l}\ c, u) \;\underset{q_{c,u,v}}{\text{————}}\; e(\mathtt{r}\ c, v)$$

The induction principle then gives us a fibered homomorphism (f, β, θ), where $f : D \to E$,

$$\beta(a, t) : f(d(a, t)) = e(a, f \circ t)$$

and $\theta(c, t, s)$ implies the commutativity of the following diagram:

$$\begin{array}{ccccc}
(p_{c,t,s})_*^{-\mapsto Y}(f(d(\mathtt{l}\ c, t))) & & \overset{\mathsf{dap}_f(p_{c,t,s})}{\text{—————————————}} & & f(d(\mathtt{r}\ c, s)) \\
\mathsf{ap}_{p(c,t,s)_*^{-\mapsto Y}}(\beta_{\mathtt{l}(c),t}) \Big| & & & & \Big| \beta_{\mathtt{r}(c),s} \\
(p_{c,t,s})_*^{-\mapsto Y}(e(\mathtt{l}\ c, f \circ t)) & \text{———} & e(\mathtt{l}\ c, f \circ t) & \underset{q_{c,f\circ t,f\circ s}}{\text{————}} & e(\mathtt{r}\ c, f \circ s)
\end{array}$$

The terms f and β form the first two components of our desired homomorphism from (D, d, p) to (E, e, q). For the last component, we note that using path induction we can express $\mathsf{dap}_f(p_{c,t,s})$ equivalently as the path

$$(p_{c,t,s})_*^{-\mapsto Y}(f(d(\mathtt{l}\ c, t))) \;\text{———}\; f(d(\mathtt{l}\ c, t)) \;\overset{\mathsf{ap}_f(p_{c,t,s})}{\text{————}}\; f(d(\mathtt{r}\ c, s))$$

Thus the outer rectangle in the following diagram commutes:

$$\begin{array}{ccccc}
(p_{c,t,s})_*^{-\mapsto Y}(f(d(\mathtt{l}\ c, t))) & \text{———} & f(d(\mathtt{l}\ c, t)) & \overset{\mathsf{ap}_f(p_{c,t,s})}{\text{————}} & f(d(\mathtt{r}\ c, s)) \\
\mathsf{ap}_{p(c,t,s)_*^{-\mapsto Y}}(\beta_{\mathtt{l}(c),t}) \Big| & A & \beta_{\mathtt{l}(c),t} \Big| & B & \Big| \beta_{\mathtt{r}(c),s} \\
(p_{c,t,s})_*^{-\mapsto Y}(e(\mathtt{l}\ c, f \circ t)) & \text{———} & e(\mathtt{l}\ c, f \circ t) & \underset{q_{c,f\circ t,f\circ s}}{\text{————}} & e(\mathtt{r}\ c, f \circ s)
\end{array}$$

Suitable path induction shows that rectangle *A* commutes; hence rectangle *B* commutes too and we are done.

C Induction implies uniqueness

In this section we want to show that

$$\mathsf{has\text{-}W\text{-}ind}_{\mathcal{U}_k}(\mathcal{X}) \to \mathsf{has\text{-}W\text{-}ind\text{-}uniq}_{\mathcal{U}_k}(\mathcal{X})$$
$$\mathsf{has\text{-}W\text{-}ind}_{\mathcal{U}_k}(\mathcal{X}) \to \mathsf{has\text{-}W\text{-}rec\text{-}uniq}_{\mathcal{U}_k}(\mathcal{X})$$

for any $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}$. We first show the former.

Fix an algebra $(D, d, p) : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}$ and assume that $\mathsf{has\text{-}W\text{-}ind}_{\mathcal{U}_k}(D, d, p)$ holds. To show that $\mathsf{has\text{-}W\text{-}ind\text{-}uniq}_{\mathcal{U}_k}(D, d, p)$ holds, fix any fibered algebra $(E, e, q) : \mathsf{W\text{-}Fib\text{-}Alg}_{\mathcal{U}_k}\ (D, d, p)$ and fibered homomorphisms $(f, \beta, \theta), (g, \gamma, \phi) : \mathsf{W\text{-}Fib\text{-}Hom}\ (D, d, p)\ (E, e, q)$. By Lem. 73, to show $(f, \beta, \theta) = (g, \gamma, \phi)$ it suffices to exhibit a fibered W-cell between (f, β, θ) and (g, γ, ϕ). To do so, we use the induction principle with an appropriate fibered algebra (E', e', q'). Defining the first component is easy: we put $E'(x) :\equiv \mathsf{Id}_{E(x)}(f(x), g(x))$, which clearly still belongs to $\mathcal{U}_k$. For the second component, we put

$$e'(a, t, u) :\equiv \beta(a, t) \cdot \mathsf{ap}_{e(a,t)}({}^{\Pi}\mathsf{E}^{=}(u)) \cdot \gamma(a, t)^{-1}$$

For the last component, we first make some definitions: for any $r : x =_D y$, $z : f(x) = g(x)$ we put

$$\mathcal{T}^1_r(z) :\equiv \big(\mathsf{dap}_f(r)\big)^{-1} \cdot \big(\mathsf{ap}_{r^E_*}(z) \cdot \mathsf{dap}_g(r)\big)$$

and for any $r : x =_D y$, $z : f(y) = g(y)$ we put

$$\mathcal{T}^2_r(z) :\equiv \big(\mathsf{dap}_f(r)\big)^{-1} \cdot \big(\mathsf{dap}_f(r) \cdot z\big)$$

Clearly, we have $\mathcal{T}^1_r(z) = r^{E'}_*(z)$ and $\mathcal{T}^2_r(z) = z$. We can thus define $q'(c, t, s, u, v)$ to be the path in Fig. 7 b).

The induction principle then gives us a fibered homomorphism (α, η, ψ), where $\alpha : f \sim g$,

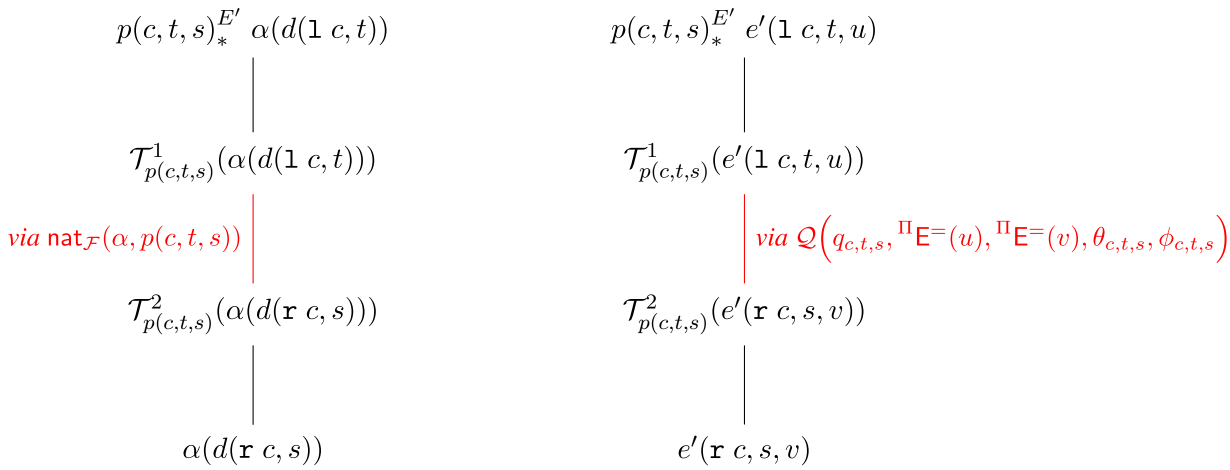
$$\eta(a, t) : \alpha(d(a, t)) = e'(a, t, \alpha \circ t)$$

and $\psi(c, t, s)$ implies the commutativity of the diagram in Fig. 8. The terms α and η form the first two components of the desired fibered W-cell (f, β, θ) and (g, γ, ϕ). For the last component, we note that the term $\mathsf{dap}_\alpha(p(c, t, s))$ can be expressed equivalently as the path in Fig. 7 a).

Thus, the outer rectangle in Fig. 9 commutes. We can easily show that the rectangles *A* and *C* commute; thus *B* commutes as well. It is easy to see that this implies the commutativity of the diagram in Fig. 4 as desired.

The non-dependent case proceeds by an entirely analogous argument, further simplified by the fact that we no longer need to transport along the fibers of the codomain type E.

Remark: With some effort, we could obtain the non-dependent case from the result we have just proved. However, due to the presence of superfluous transports, it is much simpler to establish the non-dependent result directly, following the same methodology.



a) **b)**

Figure 6: Paths $\mathsf{dap}_\alpha(p(c,t,s))$ and $q'(c,t,s,u,v)$

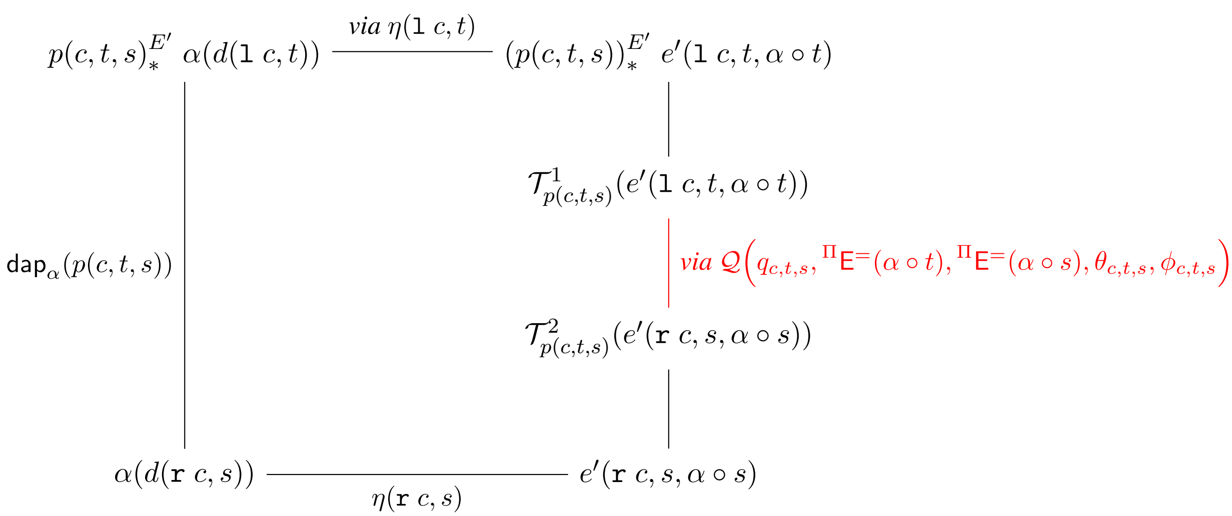


Figure 7: Commuting diagram implied by $\psi(c,t,s)$

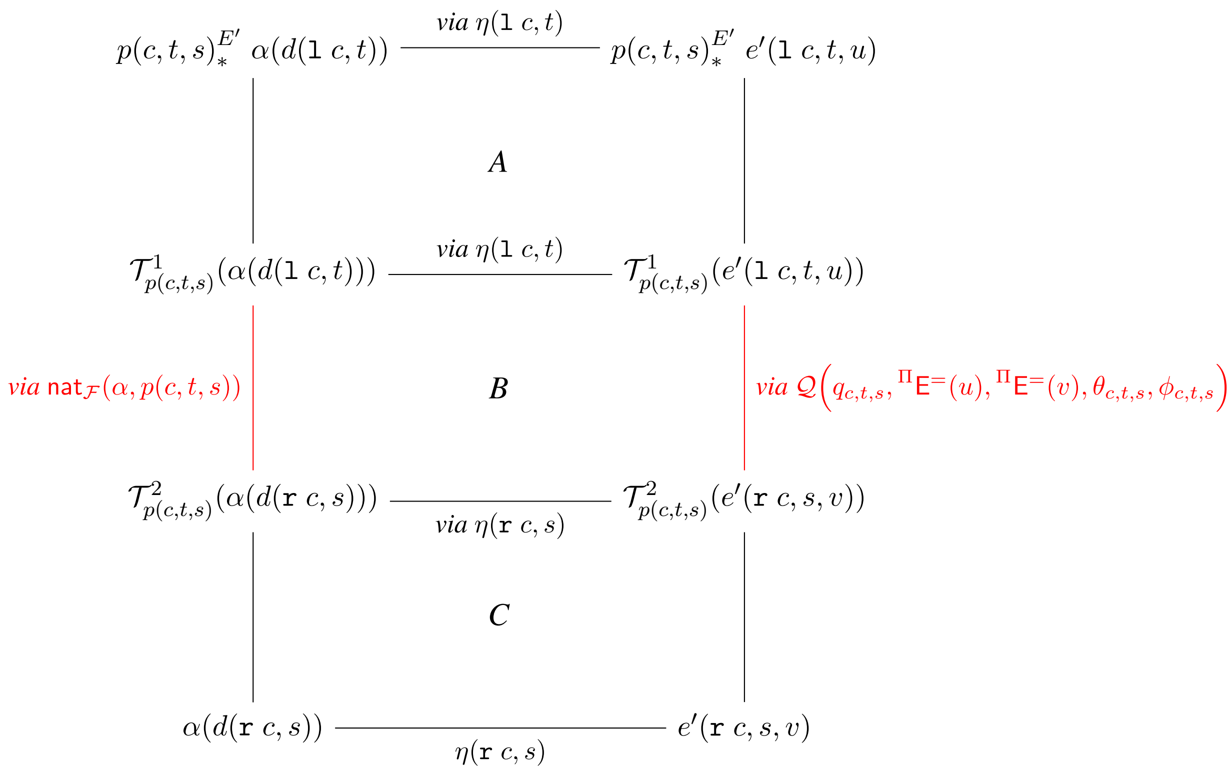


Figure 8: Diagram from Fig. 8 after expansion

D Recursion plus uniqueness implies induction

In this section we want to show that

$$\mathsf{has\text{-}W\text{-}rec}_{\mathcal{U}_k}(\mathcal{X}) \times \mathsf{has\text{-}W\text{-}rec\text{-}uniq}_{\mathcal{U}_k}(\mathcal{X}) \to \mathsf{has\text{-}W\text{-}ind}_{\mathcal{U}_k}(\mathcal{X})$$

for any $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}$. Fix an algebra $(D, d, p) : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}$ and assume that $\mathsf{has\text{-}W\text{-}rec}_{\mathcal{U}_k}(D, d, p)$ and $\mathsf{has\text{-}W\text{-}rec\text{-}uniq}_{\mathcal{U}_k}(D, d, p)$ hold. To show that $\mathsf{has\text{-}W\text{-}ind}_{\mathcal{U}_k}(D, d, p)$ holds, fix any fibered algebra $(E, e, q) : \mathsf{W\text{-}Fib\text{-}Alg}_{\mathcal{U}_k}\ (D, d, p)$. In order to apply the recursion principle, we need to turn this into a non-fibered algebra (E', e', q'). The first component is easy: the only reasonable choice we have is to put $E' :\equiv \Sigma_{x:D} E(x)$; we note that since $D : \mathcal{U}_j$, $E : D \to \mathcal{U}_k$, and $j \leq k$, we indeed have $\Sigma_{x:D} E(x) : \mathcal{U}_k$ as needed. For the second component, we put

$$e'(a, u) :\equiv \Big(d\big(a, \pi_1 \circ u\big), e\big(a, \pi_1 \circ u, \pi_2 \circ u\big)\Big)$$

For the last component, we define $q'(c, u, v)$ to be the path

$$\Big(d\big(\mathtt{l}\ c, \pi_1 \circ u\big), e\big(\mathtt{l}\ c, \pi_1 \circ u, \pi_2 \circ u\big)\Big)$$

$$\Big| \ {}^{\Sigma}\mathsf{E}^{=}\big(p(c, \pi_1 \circ u, \pi_1 \circ v), q(c, \pi_1 \circ u, \pi_1 \circ v, \pi_2 \circ u, \pi_2 \circ v)\big)$$

$$\Big(d\big(\mathtt{r}\ c, \pi_1 \circ v\big), e\big(\mathtt{r}\ c, \pi_1 \circ v, \pi_2 \circ v\big)\Big)$$

The recursion principle then gives us a homomorphism (f, β, θ), where $f : D \to \Sigma_{x:D} E(x)$,

$$\begin{aligned} \beta(a, t) &: f(d(a, t)) = e'(a, f \circ t) \\ \theta(c, t, s) &: \mathsf{ap}_f(p(c, t, s)) = \beta(\mathtt{l}\ c, t) \cdot q'(c, f \circ t, f \circ s) \cdot \beta(\mathtt{r}\ c, s)^{-1} \end{aligned}$$

We now want to show that the function $\pi_1 \circ f : D \to D$ is in fact the identity on D (up to a homotopy, of course). We can do this by endowing both of the functions $\pi_1 \circ f$ and id_D with a homomorphism structure on the algebra (D, d, p); by the recursion uniqueness principle it will follow that these homomorphisms are equal, and in particular they are equal as maps.

We have a homomorphism $\big(\mathsf{id}_D; \delta; c, t, s \mapsto \mathbf{I}_{\Box}^{-2}(\varphi_{c,t,s})\big) : \mathsf{W\text{-}Hom}\ (D, d, p)\ (D, d, p)$ where $\delta(a, t) :\equiv \mathsf{refl}_D(d(a, t))$ and $\varphi(c, t, s)$ is the obvious path

$$\mathsf{refl} \cdot \mathsf{ap}_{\mathsf{id}}(p(c, t, s))$$

$$\Big|$$

$$p(c, t, s) \cdot \mathsf{refl}$$

We also have a homomorphism $\big(\pi_1 \circ f; \gamma; c, t, s \mapsto \mathbf{I}_{\Box}^{-1}(\phi_{c,t,s}^{-1})\big) : \mathsf{W\text{-}Hom}\ (D, d, p)\ (D, d, p)$ where $\gamma(a, t) :\equiv \pi_1({}^{=}\mathsf{E}^{\Sigma}(\beta(a, t)))$ and $\phi(c, t, s)$ is the path in Fig. 10.

By the recursion uniqueness rule, the homomorphisms $\big(\pi_1 \circ f; \gamma; c, t, s \mapsto \mathbf{I}_{\Box}^{-1}(\phi_{c,t,s}^{-1})\big)$ and $\big(\mathsf{id}_D; \delta; c, t, s \mapsto \mathbf{I}_{\Box}^{-2}(\varphi_{c,t,s})\big)$ are equal. By Lem. 72 there exists a W-cell (α, η, ψ) between them, where $\alpha : \pi_1 \circ f \sim \mathsf{id}_D$,

$$\eta(a,t) : \alpha(d(a,t)) = \pi_1({}^{=}\mathsf{E}^{\Sigma}(\beta(a,t))) \centerdot \mathsf{ap}_{d(a)}({}^{\Pi}\mathsf{E}^{=}(\alpha \circ t)) \centerdot \mathsf{refl}_D(d(a,t))$$

and $\psi(c,t,s)$ implies the commutativity of diagram in Fig. 11.

We observe that the path $\mathcal{P}\big(p(c), {}^{\Pi}\mathsf{E}^{=}(\alpha \circ t), {}^{\Pi}\mathsf{E}^{=}(\alpha \circ s), \mathbf{I}_{\Box}^{-1}(\phi_{c,t,s}^{-1}), \mathbf{I}_{\Box}^{-2}(\varphi_{c,t,s})\big)$, shown in Fig. 12 for reference, can be expressed equivalently as the path in Fig. 13.

Now we observe that for any $a : A$, path $r : t =_{B(a)\to D} s$, and $u : \Pi_{b:B(a)} E(t\ b)$, we have a higher path

$$\begin{array}{c} \big(\mathsf{ap}_{d(a)}(r)\big)_*^E\ e(a,t,u) \\ \Big\vert\ \epsilon(a,r,u) \\ e\Big(a, s, \big({}^{=}\mathsf{E}^{\Pi}(r) \circ_{\mathcal{H}} u\big)\Big) \end{array}$$

defined by path induction on r.

The desired fibered homomorphism (f_D, β_D, θ_D) : W-Fib-Hom $(D, d, p)\ (E, e, q)$ can now be constructed as follows. We let $f_D(x) :\equiv \alpha \circ_{\mathcal{H}} (\pi_2 \circ u)$ and $\beta_D(a,t)$ be the path in Fig. 14. To construct θ_D, we first establish the following lemma.

Lemma 83. *Let the following data be given:*

- $a_1, a_2 : D$ *and* $u : a_1 = a_2$
- $d_1, d_2 : D$ *and* $v : d_1 = d_2$
- $w_1 : d_1 = a_1$ *and* $w_2 : d_2 = a_2$
- $e_1 : E(d_1), e_1' : E(a_1), e_1'' : E(a_1)$ *and* $e_2 : E(d_2), e_2' : E(a_2), e_2'' : E(a_2)$
- $z : v_*^E(e_1) = e_2$ *and* $z' : u_*^E(e_1') = e_2'$ *and* $z'' : u_*^E(e_1'') = e_2''$
- $\epsilon_1 : (w_1)_*^E(e_1) = e_1'$ *and* $\epsilon_2 : (w_2)_*^E(e_2) = e_2'$
- $\kappa_1 : e_1' = e_1''$ *and* $\kappa_2 : e_2' = e_2''$
- $\beta_1 : f(a_1) = (d_1, e_1)$ *and* $\beta_2 : f(a_2) = (d_2, e_2)$
- $\eta_1 : \alpha(a_1) = \pi_1({}^{=}\mathsf{E}^{\Sigma}(\beta_1)) \centerdot w_1 \centerdot \mathsf{refl}(a_1)$ *and* $\eta_2 : \alpha(a_2) = \pi_1({}^{=}\mathsf{E}^{\Sigma}(\beta_2)) \centerdot w_2 \centerdot \mathsf{refl}(a_2)$
- $\Theta : v \centerdot w_2 = w_1 \centerdot u$
- $\Phi : \mathsf{ap}_f(u) \centerdot \beta_2 = \beta_1 \centerdot {}^{\Sigma}\mathsf{E}^{=}(v, z)$

such that the diagrams in Fig. 15 commute. Then the diagram below commutes:

$$
\begin{array}{ccc}
u^E_*\big(\alpha(a_1)^E_*\ \pi_2(f(a_1))\big) & \overset{\mathsf{dap}_{\alpha \circ_{\mathcal{H}} (\pi_2 \circ f)}(u)}{\rule[0.5ex]{8em}{0.4pt}} & \alpha(a_2)^E_*\ \pi_2(f(a_2)) \\
\mathsf{ap}_{u^E_*}(\beta^D_1) \Big| & & \Big| \beta^D_2 \\
u^E_*(e''_1) & \underset{z''}{\rule[0.5ex]{8em}{0.4pt}} & e''_2
\end{array}
$$

where β^D_1 and β^D_2 are the paths in Fig. 16 a) and b) respectively.

Proof. We proceed by path induction on u. The diagrams in Fig. 15 reduce to those in Fig. 17. It is not hard to see that the commutativity of the diagram in Fig. 17 c) is equivalent to the commutativity of the diagram in Fig. 18.

The path $\mathsf{dap}_{\alpha \circ_{\mathcal{H}} (\pi_2 \circ f)}(u) \cdot \beta^D_2$ reduces to the path in Fig. 19. After a simple expansion we obtain the path in Fig. 20.

The path $\mathsf{ap}_{u^E_*}(\beta^D_1) \cdot z''$ reduces to the path in Fig. 21. Using commutativity of the diagram in Fig. 17 a), we obtain the path in Fig. 22. A further rearrangement yields the path in Fig. 23. Using the commutativity of the diagram in Fig. 17 b), we obtain the path in Fig. 24. After successive rearrangements, we get the diagrams in Fig. 25, 26, and 27.

Finally, the commutativity of the diagram in Fig. 17 c) implies that the paths in Fig. 20 and 27 are equal as desired. □

At last, to construct $\theta_D(c, t, s)$ we need to show that the following diagram commutes:

$$
\begin{array}{ccc}
p(c,t,s)^E_*\big(f_D(d(\mathtt{l}\ c, t))\big) & \overset{\mathsf{dap}_{f_D}(p(c,t,s))}{\rule[0.5ex]{10em}{0.4pt}} & f_D(d(\mathtt{r}\ c, s)) \\
\mathsf{ap}_{p(c,t,s)^E_*}(\beta_D(\mathtt{l}\ c, t)) \Big| & & \Big| \beta_D(\mathtt{r}\ c, s) \\
p(c,t,s)^E_*\big(e(\mathtt{l}\ c, t, f_D \circ t)\big) & \underset{q(c,t,s,f_D \circ t, f_D \circ s)}{\rule[0.5ex]{10em}{0.4pt}} & e(\mathtt{r}\ c, s, f_D \circ s)
\end{array}
$$

We use Lem. 83 with the following data:

- $a_1 :\equiv d(\mathtt{l}\ c, t)$, $a_2 :\equiv d(\mathtt{r}\ c, s)$, $u :\equiv p(c, t, s)$
- $d_1 :\equiv d(\mathtt{l}\ c, \pi_1 \circ f \circ t)$, $d_2 :\equiv d(\mathtt{r}\ c, \pi_1 \circ f \circ s)$, $v :\equiv p(c, \pi_1 \circ f \circ t, \pi_1 \circ f \circ s)$
- $w_1 :\equiv \mathsf{ap}_{d(\mathtt{l}\ c)}({}^{\Pi}\mathsf{E}^{=}(\alpha \circ t))$, $w_2 :\equiv \mathsf{ap}_{d(\mathtt{r}\ c)}({}^{\Pi}\mathsf{E}^{=}(\alpha \circ s))$
- $e_1 :\equiv e\big(\mathtt{l}\ c, \pi_1 \circ f \circ t, \pi_2 \circ f \circ t\big)$, $e_2 :\equiv e\big(\mathtt{r}\ c, \pi_1 \circ f \circ s, \pi_2 \circ f \circ s\big)$
- $e'_1 :\equiv e\Big(\mathtt{l}\ c, t, \Big({}^{=}\mathsf{E}^{\Pi}({}^{\Pi}\mathsf{E}^{=}(\alpha \circ t)) \circ_{\mathcal{H}} (\pi_2 \circ f \circ t)\Big)\Big)$

- $e_2' :\equiv e\Big(\mathtt{r}\ c, s, \Big({}^{=}\mathsf{E}^{\Pi}({}^{\Pi}\mathsf{E}^{=}(\alpha \circ s)) \circ_{\mathcal{H}} (\pi_2 \circ f \circ s)\Big)\Big)$
- $e_1'' :\equiv e\Big(\mathtt{l}\ c, t, \Big((\alpha \circ t) \circ_{\mathcal{H}} (\pi_2 \circ f \circ t)\Big)\Big)$, $e_2'' :\equiv e\Big(\mathtt{r}\ c, s, \Big((\alpha \circ s) \circ_{\mathcal{H}} (\pi_2 \circ f \circ s)\Big)\Big)$
- $z :\equiv q\Big(c, \pi_1 \circ f \circ t, \pi_1 \circ f \circ s, \pi_2 \circ f \circ t, \pi_2 \circ f \circ s\Big)$
- $z' :\equiv q\Big(c, t, s, \Big({}^{=}\mathsf{E}^{\Pi}({}^{\Pi}\mathsf{E}^{=}(\alpha \circ t)) \circ_{\mathcal{H}} (\pi_2 \circ f \circ t)\Big), \Big({}^{=}\mathsf{E}^{\Pi}({}^{\Pi}\mathsf{E}^{=}(\alpha \circ s)) \circ_{\mathcal{H}} (\pi_2 \circ f \circ s)\Big)\Big)$
- $z'' :\equiv q\Big(c, t, s, \Big((\alpha \circ t) \circ_{\mathcal{H}} (\pi_2 \circ f \circ t)\Big), \Big((\alpha \circ s) \circ_{\mathcal{H}} (\pi_2 \circ f \circ s)\Big)\Big)$
- $\epsilon_1 :\equiv \epsilon\big(\mathtt{l}\ c, {}^{\Pi}\mathsf{E}^{=}(\alpha \circ t), \pi_2 \circ f \circ t\big)$, $\epsilon_2 :\equiv \epsilon\big(\mathtt{r}\ c, {}^{\Pi}\mathsf{E}^{=}(\alpha \circ s), \pi_2 \circ f \circ s\big)$
- $\kappa_1 :\equiv \mathsf{ap}_{e(\mathtt{l}\ c,t)}(r_t)$, $\kappa_2 :\equiv \mathsf{ap}_{e(\mathtt{r}\ c,s)}(r_s)$ where r_t and r_s are the obvious paths below:

$$\begin{array}{c} {}^{=}\mathsf{E}^{\Pi}({}^{\Pi}\mathsf{E}^{=}(\alpha \circ t)) \circ_{\mathcal{H}} (\pi_2 \circ f \circ t) \\ \Big| \\ (\alpha \circ t) \circ_{\mathcal{H}} (\pi_2 \circ f \circ t) \end{array} \qquad \begin{array}{c} {}^{=}\mathsf{E}^{\Pi}({}^{\Pi}\mathsf{E}^{=}(\alpha \circ s)) \circ_{\mathcal{H}} (\pi_2 \circ f \circ s) \\ \Big| \\ (\alpha \circ s) \circ_{\mathcal{H}} (\pi_2 \circ f \circ s) \end{array}$$

- $\beta_1 :\equiv \beta(\mathtt{l}\ c, t)$, $\beta_2 :\equiv \beta(\mathtt{r}\ c, s)$
- $\eta_1 :\equiv \eta(\mathtt{l}\ c, t)$, $\eta_2 :\equiv \eta(\mathtt{r}\ c, s)$
- $\Theta :\equiv \mathsf{nat}_{\mathcal{H}}\big(p(c), {}^{\Pi}\mathsf{E}^{=}(\alpha \circ t), {}^{\Pi}\mathsf{E}^{=}(\alpha \circ s)\big)^{-1}$
- $\Phi :\equiv \mathbf{I}^1_{\square}(\theta(c, t, s))$

It thus suffices to show that the diagrams in Fig. 15 commute.

a) We note that we have $\mathsf{ap}_{p(c,t,s)^E_*}(\mathsf{ap}_{e(\mathtt{l}\ c,t)}(r_t)) = \mathsf{ap}_{p(c,t,s)^E_* \circ e(\mathtt{l}\ c,t)}(r_t)$ and appeal to the higher path $\mathsf{nat}_{\mathcal{H}}\big(q(c, t, s), r_t, r_s\big)$.

b) We generalize the setting as follows. Fix variables $t_1, t_2 : B(\mathtt{l}\ c) \to D$, $s_1, s_2 : B(\mathtt{r}\ c) \to D$, $\upsilon_t : t_1 = t_2$, $\upsilon_s : s_1 = s_2$, $x : \Pi_{b:B(\mathtt{l}\ c)} E(t_1(b))$, $y : \Pi_{b:B(\mathtt{r}\ c)} E(s_1(b))$. Then the diagram in Fig. 28 commutes by easy path induction on υ_t and υ_s. A suitable instantiation of the variables yields the desired diagram.

c) This follows from Fig. 11, 12, and 13.

$$
\begin{array}{c}
\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{l}(c),t})\big) \cdot p(c, \pi_1 \circ f \circ t, \pi_1 \circ f \circ s) \\
\Big| \\
\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{l}(c),t})\big) \cdot \pi_1\big({}^{=}\mathsf{E}^{\Sigma}(q'(c, f \circ t, f \circ s))\big) \\
\Big| \\
\pi_1\Big({}^{=}\mathsf{E}^{\Sigma}\big(\beta_{\mathtt{l}(c),t} \cdot q'(c, f \circ t, f \circ s)\big)\Big) \\
\Big| \ \textit{via } \mathbf{I}^1_{\square}(\theta(c,t,s))^{-1} \\
\pi_1\Big({}^{=}\mathsf{E}^{\Sigma}\big(\mathsf{ap}_f(p(c,t,s)) \cdot \beta_{\mathtt{r}(c),s}\big)\Big) \\
\Big| \\
\pi_1\big({}^{=}\mathsf{E}^{\Sigma}\big(\mathsf{ap}_f(p(c,t,s))\big)\big) \cdot \pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{r}(c),s})\big) \\
\Big| \\
\mathsf{ap}_{\pi_1}\big(\mathsf{ap}_f(p(c,t,s))\big) \cdot \pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{r}(c),s})\big) \\
\Big| \\
\mathsf{ap}_{\pi_1 \circ f}(p(c,t,s)) \cdot \pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{r}(c),s})\big)
\end{array}
$$

Figure 9: The path $\phi(c, t, s)$

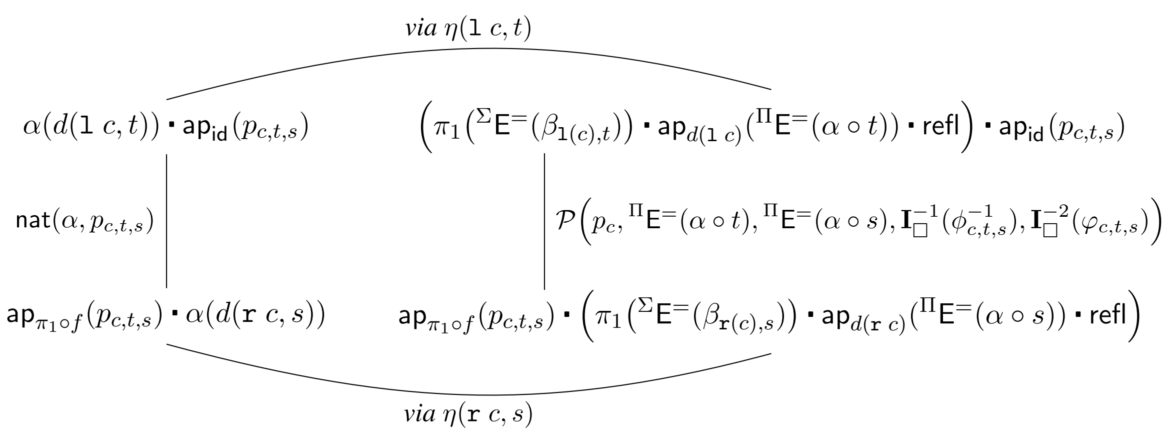


Figure 10: Commuting diagram implied by $\psi(c, t, s)$

$$\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{l}(c),t})\big) \cdot \mathsf{ap}_{d(\mathtt{l}\ c)}\big({}^{\Pi}\mathsf{E}^{=}(\alpha \circ t)\big) \cdot \mathsf{refl}\Big) \cdot \mathsf{ap}_{\mathsf{id}}(p(c,t,s))$$

$$\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{l}(c),t})\big) \cdot \mathsf{ap}_{d(\mathtt{l}\ c)}\big({}^{\Pi}\mathsf{E}^{=}(\alpha \circ t)\big)\Big) \cdot \Big(\mathsf{refl} \cdot \mathsf{ap}_{\mathsf{id}}(p(c,t,s))\Big)$$

via $\mathbf{I}^{2}_{\square}(\mathbf{I}^{-2}_{\square}(\varphi(c,t,s)))$

$$\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{l}(c),t})\big) \cdot \mathsf{ap}_{d(\mathtt{l}\ c)}\big({}^{\Pi}\mathsf{E}^{=}(\alpha \circ t)\big)\Big) \cdot \Big(p(c,t,s) \cdot \mathsf{refl}\Big)$$

$$\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{l}(c),t})\big) \cdot \Big(\mathsf{ap}_{d(\mathtt{l}\ c)}\big({}^{\Pi}\mathsf{E}^{=}(\alpha \circ t)\big) \cdot p(c,t,s)\Big) \cdot \mathsf{refl}$$

via $\mathsf{nat}_{\mathcal{H}}\big(p(c), {}^{\Pi}\mathsf{E}^{=}(\alpha \circ t), {}^{\Pi}\mathsf{E}^{=}(\alpha \circ s)\big)^{-1}$

$$\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{l}(c),t})\big) \cdot \Big(p(c, \pi_1 \circ f \circ t, \pi_1 \circ f \circ s) \cdot \mathsf{ap}_{d(\mathtt{r}\ c)}\big({}^{\Pi}\mathsf{E}^{=}(\alpha \circ s)\big)\Big) \cdot \mathsf{refl}$$

$$\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{l}(c),t})\big) \cdot p(c, \pi_1 \circ f \circ t, \pi_1 \circ f \circ s)\Big) \cdot \Big(\mathsf{ap}_{d(\mathtt{r}\ c)}\big({}^{\Pi}\mathsf{E}^{=}(\alpha \circ s)\big) \cdot \mathsf{refl}\Big)$$

via $\mathbf{I}^{1}_{\square}(\mathbf{I}^{-1}_{\square}(\phi(c,t,s)^{-1}))^{-1}$

$$\Big(\mathsf{ap}_{\pi_1 \circ f}(p(c,t,s)) \cdot \pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{r}(c),s})\big)\Big) \cdot \Big(\mathsf{ap}_{d(\mathtt{r}\ c)}\big({}^{\Pi}\mathsf{E}^{=}(\alpha \circ s)\big) \cdot \mathsf{refl}\Big)$$

$$\mathsf{ap}_{\pi_1 \circ f}(p(c,t,s)) \cdot \Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{r}(c),s})\big) \cdot \mathsf{ap}_{d(\mathtt{r}\ c)}\big({}^{\Pi}\mathsf{E}^{=}(\alpha \circ s)\big) \cdot \mathsf{refl}\Big)$$

Figure 11: The path $\mathcal{P}\Big(p(c), {}^{\Pi}\mathsf{E}^{=}(\alpha \circ t), {}^{\Pi}\mathsf{E}^{=}(\alpha \circ s), \mathbf{I}^{-1}_{\square}(\phi^{-1}_{c,t,s}), \mathbf{I}^{-2}_{\square}(\varphi_{c,t,s})\Big)$

$$\left(\pi_1\left({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{l}(c),t})\right) \cdot \mathsf{ap}_{d(\mathtt{l}\ c)}\left({}^{\Pi}\mathsf{E}^{=}(\alpha \circ t)\right) \cdot \mathsf{refl}\right) \cdot \mathsf{ap}_{\mathsf{id}}(p(c,t,s))$$

$$\left(\pi_1\left({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{l}(c),t})\right) \cdot \mathsf{ap}_{d(\mathtt{l}\ c)}\left({}^{\Pi}\mathsf{E}^{=}(\alpha \circ t)\right)\right) \cdot \mathsf{ap}_{\mathsf{id}}(p(c,t,s))$$

$$\left(\pi_1\left({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{l}(c),t})\right) \cdot \mathsf{ap}_{d(\mathtt{l}\ c)}\left({}^{\Pi}\mathsf{E}^{=}(\alpha \circ t)\right)\right) \cdot p(c,t,s)$$

$$\pi_1\left({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{l}(c),t})\right) \cdot \left(\mathsf{ap}_{d(\mathtt{l}\ c)}\left({}^{\Pi}\mathsf{E}^{=}(\alpha \circ t)\right) \cdot p(c,t,s)\right)$$

via $\mathsf{nat}_{\mathcal{H}}\left(p(c), {}^{\Pi}\mathsf{E}^{=}(\alpha \circ t), {}^{\Pi}\mathsf{E}^{=}(\alpha \circ s)\right)^{-1}$

$$\pi_1\left({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{l}(c),t})\right) \cdot \left(p(c, \pi_1 \circ f \circ t, \pi_1 \circ f \circ s) \cdot \mathsf{ap}_{d(\mathtt{r}\ c)}\left({}^{\Pi}\mathsf{E}^{=}(\alpha \circ s)\right)\right)$$

$$\left(\pi_1\left({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{l}(c),t})\right) \cdot p(c, \pi_1 \circ f \circ t, \pi_1 \circ f \circ s)\right) \cdot \mathsf{ap}_{d(\mathtt{r}\ c)}\left({}^{\Pi}\mathsf{E}^{=}(\alpha \circ s)\right)$$

via $\phi(c,t,s)$

$$\left(\mathsf{ap}_{\pi_1 \circ f}(p(c,t,s)) \cdot \pi_1\left({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{r}(c),s})\right)\right) \cdot \mathsf{ap}_{d(\mathtt{r}\ c)}\left({}^{\Pi}\mathsf{E}^{=}(\alpha \circ s)\right)$$

$$\mathsf{ap}_{\pi_1 \circ f}(p(c,t,s)) \cdot \left(\pi_1\left({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{r}(c),s})\right) \cdot \mathsf{ap}_{d(\mathtt{r}\ c)}\left({}^{\Pi}\mathsf{E}^{=}(\alpha \circ s)\right)\right)$$

$$\mathsf{ap}_{\pi_1 \circ f}(p(c,t,s)) \cdot \left(\pi_1\left({}^{=}\mathsf{E}^{\Sigma}(\beta_{\mathtt{r}(c),s})\right) \cdot \mathsf{ap}_{d(\mathtt{r}\ c)}\left({}^{\Pi}\mathsf{E}^{=}(\alpha \circ s)\right) \cdot \mathsf{refl}\right)$$

Figure 12: The path from Fig. 12 after some cleanup

$$\alpha(d(a,t))_*^E\ \pi_2(f(d(a,t)))$$

via $\eta(a,t)$

$$\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_{a,t})\big) \cdot \mathsf{ap}_{d(a)}\big({}^{\Pi}\mathsf{E}^{=}(\alpha \circ t)\big) \cdot \mathsf{refl}\Big)_*^E\ \pi_2(f(d(a,t)))$$

$$\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_{a,t})\big) \cdot \mathsf{ap}_{d(a)}\big({}^{\Pi}\mathsf{E}^{=}(\alpha \circ t)\big)\Big)_*^E\ \pi_2(f(d(a,t)))$$

$$\big(\mathsf{ap}_{d(a)}({}^{\Pi}\mathsf{E}^{=}(\alpha \circ t))\big)_*^E\ \Big(\big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_{a,t})\big)\big)_*^E\ \pi_2(f(d(a,t)))\Big)$$

via $\pi_2\big({}^{=}\mathsf{E}^{\Sigma}(\beta_{a,t})\big)$

$$\big(\mathsf{ap}_{d(a)}({}^{\Pi}\mathsf{E}^{=}(\alpha \circ t))\big)_*^E\ e\big(a, \pi_1 \circ f \circ t, \pi_2 \circ f \circ t\big)$$

$\epsilon\big(a, {}^{\Pi}\mathsf{E}^{=}(\alpha \circ t), \pi_2 \circ f \circ t\big)$

$$e\Big(a, t, \Big({}^{=}\mathsf{E}^{\Pi}\big({}^{\Pi}\mathsf{E}^{=}(\alpha \circ t)\big) \circ_{\mathcal{H}} (\pi_2 \circ f \circ t)\Big)\Big)$$

$$e\Big(a, t, \Big((\alpha \circ t) \circ_{\mathcal{H}} (\pi_2 \circ f \circ t)\Big)\Big)$$

Figure 13: The path $\beta_D(a,t)$

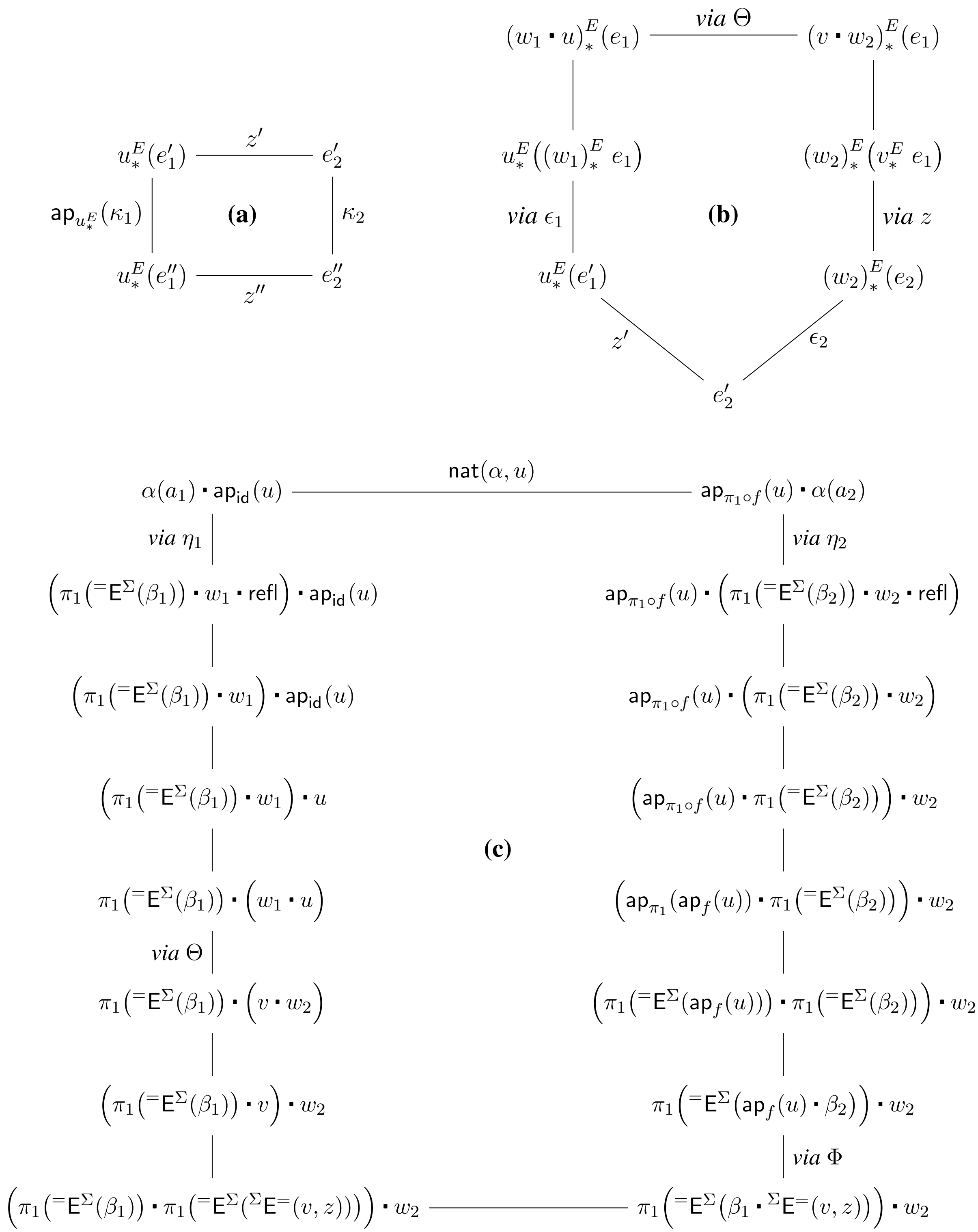


Figure 14: Hypotheses of Lem. 83

(a)

$$\alpha(a_1)^E_* \; \pi_2(f(a_1))$$

$$\Big| \; \textit{via } \eta_1$$

$$\Big(\pi_1\big({=}\mathsf{E}^\Sigma(\beta_1)\big) \cdot w_1 \cdot \mathsf{refl}\Big)^E_* \; \pi_2(f(a_1))$$

$$\Big|$$

$$\Big(\pi_1\big({=}\mathsf{E}^\Sigma(\beta_1)\big) \cdot w_1\Big)^E_* \; \pi_2(f(a_1))$$

$$\Big|$$

$$(w_1)^E_* \; \Big(\big(\pi_1\big({=}\mathsf{E}^\Sigma(\beta_1)\big)\big)^E_* \; \pi_2(f(a_1))\Big)$$

$$\Big| \; \textit{via } \pi_2\big({=}\mathsf{E}^\Sigma(\beta_1)\big)$$

$$(w_1)^E_* \; e_1$$

$$\Big| \; \epsilon_1$$

$$e'_1$$

$$\Big| \; \kappa_1$$

$$e''_1$$

(b)

$$\alpha(a_2)^E_* \; \pi_2(f(a_2))$$

$$\Big| \; \textit{via } \eta_2$$

$$\Big(\pi_1\big({=}\mathsf{E}^\Sigma(\beta_2)\big) \cdot w_2 \cdot \mathsf{refl}\Big)^E_* \; \pi_2(f(a_2))$$

$$\Big|$$

$$\Big(\pi_1\big({=}\mathsf{E}^\Sigma(\beta_2)\big) \cdot w_2\Big)^E_* \; \pi_2(f(a_2))$$

$$\Big|$$

$$(w_2)^E_* \; \Big(\big(\pi_1\big({=}\mathsf{E}^\Sigma(\beta_2)\big)\big)^E_* \; \pi_2(f(a_2))\Big)$$

$$\Big| \; \textit{via } \pi_2\big({=}\mathsf{E}^\Sigma(\beta_2)\big)$$

$$(w_2)^E_* \; e_2$$

$$\Big| \; \epsilon_2$$

$$e'_2$$

$$\Big| \; \kappa_2$$

$$e''_2$$

Figure 15: Paths β_1^D and β_2^D for Lem. 83

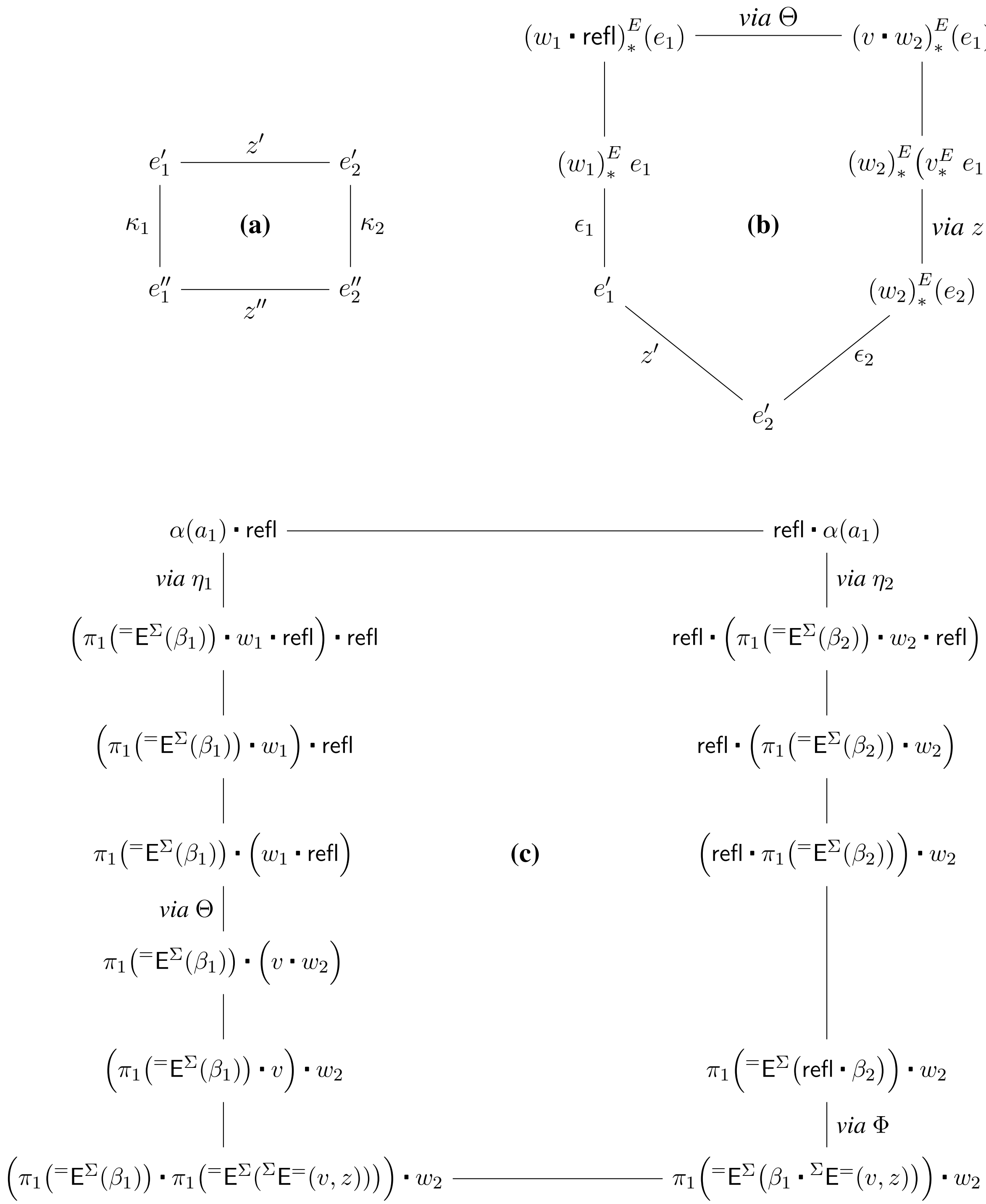


Figure 16: Hypotheses of Lem. 83 after path induction

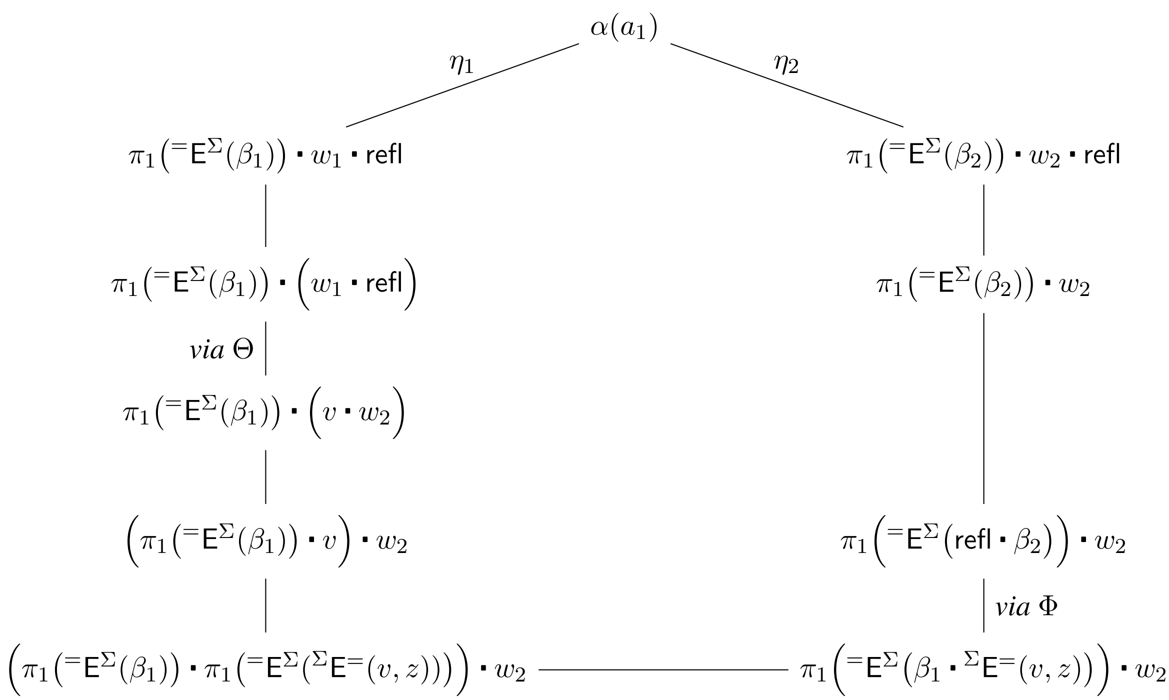


Figure 17: Diagram from Fig. 17 c) after further reduction

$$\alpha(a_1)^E_* \; \pi_2(f(a_1))$$

$$\Big| \; \textit{via } \eta_2$$

$$\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_2)\big) \cdot w_2 \cdot \mathsf{refl}\Big)^E_* \; \pi_2(f(a_1))$$

$$\Big|$$

$$\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_2)\big) \cdot w_2\Big)^E_* \; \pi_2(f(a_1))$$

$$\Big|$$

$$(w_2)^E_* \; \Big(\big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_2)\big)\big)^E_* \; \pi_2(f(a_1))\Big)$$

$$\Big| \; \textit{via } \pi_2\big({}^{=}\mathsf{E}^{\Sigma}(\beta_2)\big)$$

$$(w_2)^E_* \; e_2$$

$$\Big| \; \epsilon_2$$

$$e_2'$$

$$\Big| \; \kappa_2$$

$$e_2''$$

Figure 18: Path $\mathsf{dap}_{\alpha \circ_{\mathcal{H}} (\pi_2 \circ f)}(u) \cdot \beta_2^D$ after path induction

$$\alpha(a_1)^E_* \; \pi_2(f(a_1))$$

via η_2

$$\left(\pi_1\left({}^{=}\mathsf{E}^{\Sigma}(\beta_2)\right) \cdot w_2 \cdot \mathsf{refl}\right)^E_* \; \pi_2(f(a_1))$$

$$\left(\pi_1\left({}^{=}\mathsf{E}^{\Sigma}(\beta_2)\right) \cdot w_2\right)^E_* \; \pi_2(f(a_1))$$

$$\left(\pi_1\left({}^{=}\mathsf{E}^{\Sigma}(\mathsf{refl} \cdot \beta_2)\right) \cdot w_2\right)^E_* \; \pi_2(f(a_1))$$

via Φ

$$\left(\pi_1\left({}^{=}\mathsf{E}^{\Sigma}\left(\beta_1 \cdot {}^{\Sigma}\mathsf{E}^{=}(v, z)\right)\right) \cdot w_2\right)^E_* \; \pi_2(f(a_1))$$

$$(w_2)^E_* \; \left(\left(\pi_1\left({}^{=}\mathsf{E}^{\Sigma}\left(\beta_1 \cdot {}^{\Sigma}\mathsf{E}^{=}(v, z)\right)\right)\right)^E_* \; \pi_2(f(a_1))\right)$$

via $\pi_2\left({}^{=}\mathsf{E}^{\Sigma}\left(\beta_1 \cdot {}^{\Sigma}\mathsf{E}^{=}(v, z)\right)\right)$

$$(w_2)^E_* \; e_2$$

ϵ_2

$$e_2'$$

κ_2

$$e_2''$$

Figure 19: Path from Fig. 19 after expansion

$$\alpha(a_1)^E_* \; \pi_2(f(a_1))$$

via η_1

$$\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big) \bullet w_1 \bullet \mathsf{refl}\Big)^E_* \; \pi_2(f(a_1))$$

$$\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big) \bullet w_1\Big)^E_* \; \pi_2(f(a_1))$$

$$(w_1)^E_* \; \Big(\big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big)\big)^E_* \; \pi_2(f(a_1))\Big)$$

via $\pi_2\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big)$

$$(w_1)^E_* \; e_1$$

ϵ_1

$$e'_1$$

κ_1

$$e''_1$$

z''

$$e''_2$$

Figure 20: Path $\mathsf{ap}_{u^E_*}(\beta^D_1) \bullet z''$ after path induction

$$\alpha(a_1)_*^E\ \pi_2(f(a_1))$$

via η_1

$$\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big) \cdot w_1 \cdot \mathsf{refl}\Big)_*^E\ \pi_2(f(a_1))$$

$$\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big) \cdot w_1\Big)_*^E\ \pi_2(f(a_1))$$

$$(w_1)_*^E\ \Big(\big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big)\big)_*^E\ \pi_2(f(a_1))\Big)$$

via $\pi_2\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big)$

$$(w_1)_*^E\ e_1$$

ϵ_1

$$e_1'$$

z'

$$e_2'$$

κ_2

$$e_2''$$

Figure 21: Path from Fig. 21 after utilizing the diagram in Fig. 17 a)

$$\alpha(a_1)^E_* \; \pi_2(f(a_1))$$

$$\Big| \; \textit{via } \eta_1$$

$$\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big) \cdot w_1 \cdot \mathsf{refl}\Big)^E_* \; \pi_2(f(a_1))$$

$$\Big|$$

$$\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big) \cdot \big(w_1 \cdot \mathsf{refl}\big)\Big)^E_* \; \pi_2(f(a_1))$$

$$\Big|$$

$$\big(w_1 \cdot \mathsf{refl}\big)^E_* \; \Big(\big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big)\big)^E_* \; \pi_2(f(a_1))\Big)$$

$$\Big| \; \textit{via } \pi_2\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big)$$

$$\big(w_1 \cdot \mathsf{refl}\big)^E_* \; e_1$$

$$\Big|$$

$$(w_1)^E_* \; e_1$$

$$\Big| \; \epsilon_1$$

$$e_1'$$

$$\Big| \; z'$$

$$e_2'$$

$$\Big| \; \kappa_2$$

$$e_2''$$

Figure 22: Path from Fig. 22 after further rearrangement

$$\alpha(a_1)^E_* \; \pi_2(f(a_1))$$

$\big|$ *via* η_1

$$\left(\pi_1\left({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\right) \cdot w_1 \cdot \mathsf{refl}\right)^E_* \; \pi_2(f(a_1))$$

$\big|$

$$\left(\pi_1\left({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\right) \cdot \left(w_1 \cdot \mathsf{refl}\right)\right)^E_* \; \pi_2(f(a_1))$$

$\big|$

$$\left(w_1 \cdot \mathsf{refl}\right)^E_* \; \left(\left(\pi_1\left({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\right)\right)^E_* \; \pi_2(f(a_1))\right)$$

$\big|$ *via* $\pi_2\left({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\right)$

$$(w_1 \cdot \mathsf{refl})^E_* \; e_1$$

$\big|$ *via* Θ

$$(v \cdot w_2)^E_* \; e_1$$

$\big|$

$$(w_2)^E_* \; \left(v^E_* \; e_1\right)$$

$\big|$ *via* z

$$(w_2)^E_* \; e_2$$

$\big|$ ϵ_2

$$e_2'$$

$\big|$ κ_2

$$e_2''$$

Figure 23: Path from Fig. 23 after utilizing the diagram in Fig. 17 b)

$$\alpha(a_1)_*^E\ \pi_2(f(a_1))$$

via η_1

$$\left(\pi_1\left({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\right) \cdot w_1 \cdot \mathsf{refl}\right)_*^E\ \pi_2(f(a_1))$$

$$\left(\pi_1\left({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\right) \cdot \left(w_1 \cdot \mathsf{refl}\right)\right)_*^E\ \pi_2(f(a_1))$$

via Θ

$$\left(\pi_1\left({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\right) \cdot \left(v \cdot w_2\right)\right)_*^E\ \pi_2(f(a_1))$$

$$\left(\left(\pi_1\left({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\right) \cdot v\right) \cdot w_2\right)_*^E\ \pi_2(f(a_1))$$

$$(w_2)_*^E\ \left(v_*^E\ \left(\left(\pi_1\left({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\right)\right)_*^E\ \pi_2(f(a_1))\right)\right)$$

via $\pi_2\left({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\right)$

$$(w_2)_*^E\ (v_*^E\ e_1)$$

via z

$$(w_2)_*^E\ e_2$$

ϵ_2

$$e_2'$$

κ_2

$$e_2''$$

Figure 24: Path from Fig. 24 after rearrangement

$$\alpha(a_1)^E_* \; \pi_2(f(a_1))$$

$$\Big\downarrow \textit{via } \eta_1$$

$$\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big) \cdot w_1 \cdot \mathsf{refl}\Big)^E_* \; \pi_2(f(a_1))$$

$$\Big|$$

$$\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big) \cdot \big(w_1 \cdot \mathsf{refl}\big)\Big)^E_* \; \pi_2(f(a_1))$$

$$\Big|\; \textit{via } \Theta$$

$$\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big) \cdot \big(v \cdot w_2\big)\Big)^E_* \; \pi_2(f(a_1))$$

$$\Big|$$

$$\Big(\big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big) \cdot v\big) \cdot w_2\Big)^E_* \; \pi_2(f(a_1))$$

$$\Big|$$

$$\Big(\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big) \cdot \pi_1\big({}^{=}\mathsf{E}^{\Sigma}\big({}^{\Sigma}\mathsf{E}^{=}(v,z)\big)\big)\Big) \cdot w_2\Big)^E_* \; \pi_2(f(a_1))$$

$$\Big|$$

$$(w_2)^E_* \; \Big(\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}\big({}^{\Sigma}\mathsf{E}^{=}(v,z)\big)\big)\Big)^E_* \; \Big(\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big)\Big)^E_* \; \pi_2(f(a_1))\Big)\Big)$$

$$\Big|\; \textit{via } \pi_2\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big)$$

$$(w_2)^E_* \; \Big(\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}\big({}^{\Sigma}\mathsf{E}^{=}(v,z)\big)\big)\Big)^E_* \; e_1\Big)$$

$$\Big|\; \textit{via } \pi_2\big({}^{=}\mathsf{E}^{\Sigma}\big({}^{\Sigma}\mathsf{E}^{=}(v,z)\big)\big)$$

$$(w_2)^E_* \; e_2$$

$$\Big|\; \epsilon_2$$

$$e_2'$$

$$\Big|\; \kappa_2$$

$$e_2''$$

Figure 25: Path from Fig. 25 after further rearrangement

$$\alpha(a_1)^E_* \; \pi_2(f(a_1))$$

$$\Big| \; \textit{via } \eta_1$$

$$\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big) \cdot w_1 \cdot \mathsf{refl}\Big)^E_* \; \pi_2(f(a_1))$$

$$\Big|$$

$$\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big) \cdot (w_1 \cdot \mathsf{refl})\Big)^E_* \; \pi_2(f(a_1))$$

$$\Big| \; \textit{via } \Theta$$

$$\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big) \cdot (v \cdot w_2)\Big)^E_* \; \pi_2(f(a_1))$$

$$\Big|$$

$$\Big(\big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big) \cdot v\big) \cdot w_2\Big)^E_* \; \pi_2(f(a_1))$$

$$\Big|$$

$$\Big(\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}(\beta_1)\big) \cdot \pi_1\big({}^{=}\mathsf{E}^{\Sigma}\big({}^{\Sigma}\mathsf{E}^{=}(v,z)\big)\big)\Big) \cdot w_2\Big)^E_* \; \pi_2(f(a_1))$$

$$\Big|$$

$$\Big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}\big(\beta_1 \cdot {}^{\Sigma}\mathsf{E}^{=}(v,z)\big)\big) \cdot w_2\Big)^E_* \; \pi_2(f(a_1))$$

$$\Big|$$

$$(w_2)^E_* \; \Big(\big(\pi_1\big({}^{=}\mathsf{E}^{\Sigma}\big(\beta_1 \cdot {}^{\Sigma}\mathsf{E}^{=}(v,z)\big)\big)\big)^E_* \; \pi_2(f(a_1))\Big)$$

$$\Big| \; \textit{via } \pi_2\big({}^{=}\mathsf{E}^{\Sigma}\big(\beta_1 \cdot {}^{\Sigma}\mathsf{E}^{=}(v,z)\big)\big)$$

$$(w_2)^E_* \; e_2$$

$$\Big| \; \epsilon_2$$

$$e_2'$$

$$\Big| \; \kappa_2$$

$$e_2''$$

Figure 26: Path from Fig. 26 after further rearrangement

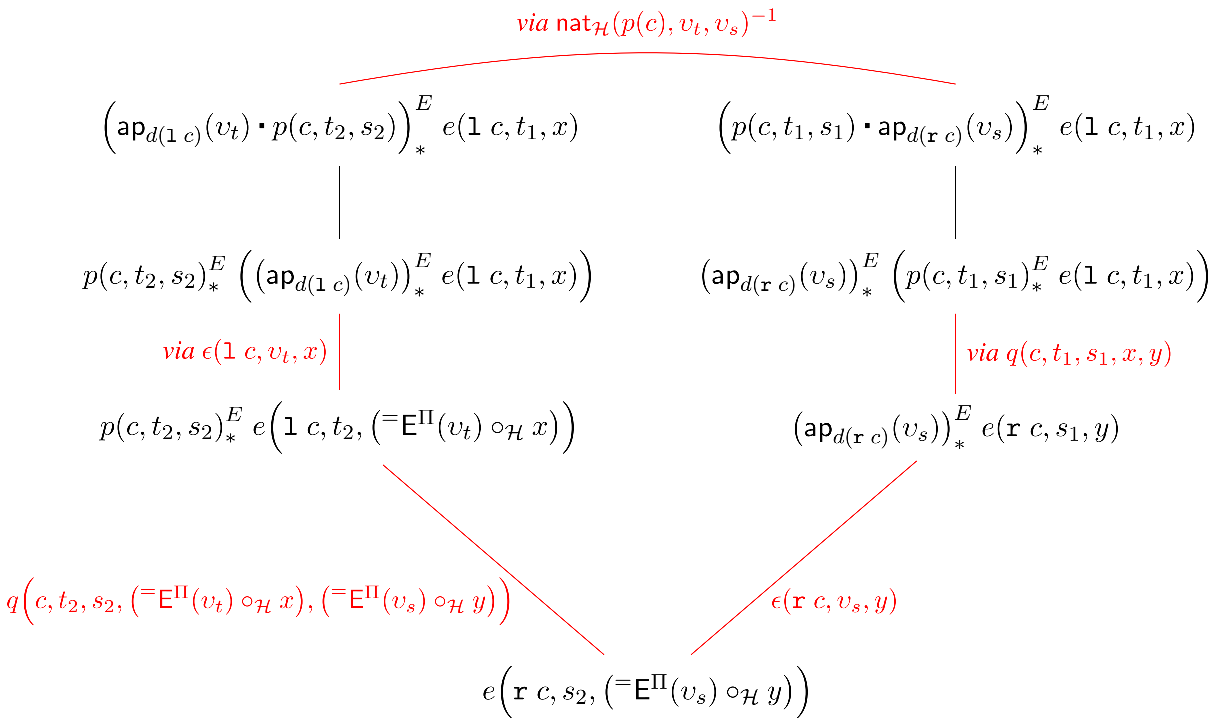


Figure 27: Instantiation of the diagram in Fig. 15 b)