A

Inductive Types in Homotopy Type Theory

STEVE AWODEY, Carnegie Mellon University
NICOLA GAMBINO, Institute for Advanced Study
KRISTINA SOJAKOVA, Carnegie Mellon University

Homotopy type theory is an interpretation of Martin-Löf's constructive type theory into abstract homotopy theory. There results a link between constructive mathematics and algebraic topology, providing topological semantics for intensional systems of type theory as well as a computational approach to algebraic topology via type theory-based proof assistants such as Coq.

The present work investigates inductive types in this setting. Modified rules for inductive types, including types of well-founded trees, or W-types, are presented, and the basic homotopical semantics of such types are determined. Proofs of all results have been formally verified by the Coq proof assistant, and the proof scripts for this verification form an essential component of this research.

Categories and Subject Descriptors: F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic

General Terms: Theory, Verification

Additional Key Words and Phrases: Homotopy type theory, homotopy-initial algebras, inductive types

ACM Reference Format:
Steve Awodey, Nicola Gambino, and Kristina Sojakova, 2014. Inductive Types in Homotopy Type Theory. *J. ACM* V, N, Article A (January YYYY), 47 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

The constructive type theories introduced by Martin-Löf are dependently-typed λ-calculi with operations for identity types $\mathsf{Id}_A(a,b)$, dependent products $(\Pi x : A)B(x)$ and dependent sums $(\Sigma x : A)B(x)$, among others [Martin-Löf 1975; 1982; 1984; Nordstrom et al. 1990; 2000]. These are related to the basic concepts of predicate logic, *viz.* equality and quantification, via the familiar propositions-as-types correspondence [Howard 1980]. The different systems introduced by Martin-Löf over the years vary greatly both in proof-theoretic strength [Griffor and Rathjen 1994] and computational properties. From the computational point of view, it is important to distinguish between the extensional systems, that have a stronger notion of equality, but for which type-checking is undecidable, and the intensional ones, that have a weaker notion of equality, but for which type-checking is decidable [Hofmann 1997]. For example, the type theory presented in [Martin-Löf 1984] is extensional, while that in [Nordstrom et al. 2000] is intensional.

The difference between the extensional and the intensional treatment of equality has a strong impact also on the properties of the various types that may be assumed

Author's addresses: S. Awodey, Philosophy Department, Carnegie Mellon University; N. Gambino, (Current address) School of Mathematics, University of Leeds; K. Sojakova, Computer Science Department, Carnegie Mellon University.

DOI:http://dx.doi.org/10.1145/0000000.0000000

in a type theory, and in particular on those of inductive types, such as the types of Booleans, natural numbers, lists and W-types [Martin-Löf 1982]. Within extensional type theories, inductive types can be characterized (up to isomorphism) as initial algebras of certain definable functors. The initiality condition translates directly into a recursion principle that expresses the existence and uniqueness of recursively-defined functions. In particular, W-types can be characterized as initial algebras of polynomial functors [Dybjer 1997; Moerdijk and Palmgren 2000]. Furthermore, within extensional type theories, W-types allow us to define a wide range of inductive types, such as the type of natural numbers and types of lists [Dybjer 1997; Gambino and Hyland 2004; Abbott et al. 2005]. Within intensional type theories, by contrast, the correspondence between inductive types and initial algebras breaks down, since it is not possible to prove the uniqueness of recursively-defined functions. Furthermore, the reduction of inductive types like the natural numbers to W-types fails [Dybjer 1997; Goguen and Luo 1993].

In the present work, we exploit insights derived from the new models of intensional type theory based on homotopy-theoretic ideas [Awodey and Warren 2009; Voevodsky 2009; van den Berg and Garner 2012] to investigate inductive types, thus contributing to the new area known as Homotopy Type Theory. Homotopical intuition justifies the assumption of a limited form of function extensionality, which, as we show, suffices to deduce uniqueness properties of recursively-defined functions up to homotopy. Building on this observation, we introduce the notions of *weak algebra homomorphism* and *homotopy-initial algebra*, which require uniqueness of homomorphisms up to homotopy. We modify the rules for W-types by replacing the definitional equality in the standard computation rule with its propositional counterpart, yielding a weak form of the corresponding inductive type. Our main result is that these new, weak W-types correspond precisely to homotopy-initial algebras of polynomial functors. Furthermore, we indicate how homotopical versions of various inductive types can be defined as special cases of the general construction in the new setting

The work presented here is motivated in part by the Univalent Foundations program formulated by Voevodsky [Voevodsky 2010b]. This ambitious program intends to provide comprehensive foundations for mathematics on the basis of homotopically-motivated type theories, with an associated computational implementation in the Coq proof assistant. The present investigation of inductive types serves as an example of this new paradigm: despite the fact that the intuitive basis lies in higher-dimensional category theory and homotopy theory, the actual development is strictly syntactic, allowing for direct formalization in Coq. Proof scripts of the definitions, results, and all necessary preliminaries are provided in a downloadable repository [Awodey et al. 2011]. An overview of these files is provided as an appendix to this paper.

The paper is organized as follows. In section 2, we describe and motivate the dependent type theory over which we will work and compare it to some other well-known systems in the literature. The basic properties of the system and its homotopical interpretation are developed to the extent required for the present purposes. Section 5.1 reviews the basic theory of W-types in extensional type theory and sketches the proof that these correspond to initial algebras of polynomial functors; there is nothing new in this section, rather it serves as a framework for the generalization that follows. Section 5.1 on intensional W-types contains the development of our new theory; it begins with a simple example, that of the type 2 of Boolean truth values, which serves to indicate the main issues involved with inductive types in the intensional setting, and our proposed solution. We then give the general notion of weak W-types, including the crucial new notion of *homotopy-initiality*, and state our main result, the equivalence between the type-theoretic rules for weak W-types and the existence of a homotopy-initial algebra of the corresponding polynomial functor. Moreover, we show how some of the difficulties with intensional W-types are remedied in the new setting by showing that the type of natural numbers can be defined as an appropriate W-type. Finally, we conclude by indicating how this work fits into the larger study of inductive types in Homotopy Type Theory and the Univalent Foundations program generally.

2. PRELIMINARIES

The general topic of Homotopy Type Theory is concerned with the study of the constructive type theories of Martin-Löf under their new interpretation into abstract homotopy theory and higher-dimensional category theory. Martin-Löf type theories are foundational systems which have been used to formalize large parts of constructive mathematics, and also for the development of high-level programming languages [Martin-Löf 1982]. They are prized for their combination of expressive strength and desirable proof-theoretic properties. One aspect of these type theories that has led to special difficulties in providing semantics is the intensional character of equality. In recent work [Awodey and Warren 2009; Voevodsky 2009; van den Berg and Garner 2012; Awodey 2010], it has emerged that the topological notion of *homotopy* provides an adequate basis for the semantics of intensionality. This extends the paradigm of computability as continuity, familiar from domain theory, beyond the simply-typed λ-calculus to dependently-typed theories involving:

(i) dependent sums $(\Sigma x\colon A)B(x)$ and dependent products $(\Pi x\colon A)B(x)$, modelled respectively by the total space and the space of sections of the fibration modelling the dependency of $B(x)$ over $x : A$;
(ii) and, crucially, including the identity type constructor $\mathsf{Id}_A(a, b)$, interpreted as the space of all *paths* in A between points a and b.

In the present work, we build on this homotopical interpretation to study inductive types, such as the natural numbers, Booleans, lists, and W-types. Within extensional type theories, W-types can be used to provide a constructive counterpart of the classical notion of a well-ordering [Martin-Löf 1984] and to uniformly define a variety of inductive types [Dybjer 1997]. However, most programming languages and proof assistants, such as Coq [Bertot and Castéran 2004], Agda [Norell 2007] and Epigram [McBride and McKinna 2004] use schematic inductive definitions [Coquand and Paulin-Mohring 1990; Paulin-Mohring 1993] rather than W-types to define inductive types. This is due in part to the practical convenience of the schematic approach, but it is also a matter of necessity; these systems are based on intensional rather than extensional type theories, and in the intensional theory the usual reductions of inductive types to W-types fail [Dybjer 1997; McBride 2010]. Nonetheless, W-types retain great importance from a theoretical perspective, since they allow us to internalize in type theory arguments about inductive types. Furthermore, as we will see in Section 5.3, a limited form of extensionality licensed by the homotopical interpretation suffices to develop the theory of W-types in a satisfactory way. In particular, we shall make use of ideas from higher category theory and homotopy theory to understand W-types as "homotopy-initial" algebras of an appropriate kind.

2.1. Extensional vs. intensional type theories

We work here with type theories that have the four standard forms of judgement

$$A : \mathsf{type}\,, \quad A = B : \mathsf{type}\,, \quad a : A\,, \quad a = b : A\,.$$

We refer to the equality relation in these judgements as *definitional equality*, which should be contrasted with the notion of *propositional equality* recalled below. Such a judgement J can be made also relative to a *context* Γ of variable declarations, a situation that we indicate by writing $\Gamma \vdash J$. When stating deduction rules we make use of standard conventions to simplify the exposition, such as omitting the mention of a context that is common to premisses and conclusions of the rule. The rules for identity types in intensional type theories are given in [Nordstrom et al. 2000, Section 5.5]. We recall them here in a slighly different, but equivalent, formulation.

— Id-formation rule.

$$\frac{A : \mathsf{type} \quad a : A \quad b : A}{\mathsf{Id}_A(a,b) : \mathsf{type}}$$

— Id-introduction rule.

$$\frac{a : A}{\mathsf{refl}(a) : \mathsf{Id}_A(a,a)}$$

— Id-elimination rule.

$$\frac{\begin{array}{c} x,y : A, u : \mathsf{Id}_A(x,y) \vdash C(x,y,u) : \mathsf{type} \\ x : A \vdash c(x) : C(x,x,\mathsf{refl}(x)) \end{array}}{x,y : A, u : \mathsf{Id}_A(x,y) \vdash \mathsf{idrec}(x,y,u,c) : C(x,y,u)}$$

— Id-computation rule.

$$\frac{\begin{array}{c} x,y : A, u : \mathsf{Id}_A(x,y) \vdash C(x,y,u) : \mathsf{type} \\ x : A \vdash c(x) : C(x,x,\mathsf{refl}(x)) \end{array}}{x : A \vdash \mathsf{idrec}(x,x,\mathsf{refl}(x),c) = c(x) : C(x,x,\mathsf{refl}(x))\,.}$$

As usual, we say that two elements $a, b : A$ are *propositionally equal* if the type $\mathsf{Id}(a,b)$ is inhabited. Most work on W-types to date (*e.g.* [Dybjer 1997; Moerdijk and Palmgren 2000; Abbott et al. 2005]) has been in the setting of extensional type theories, in which the following rule, known as the *identity reflection rule*, is also assumed:

$$\frac{p : \mathsf{Id}_A(a,b)}{a = b : A} \tag{1}$$

This rule collapses propositional equality with definitional equality, thus making the overall system somewhat simpler to work with. However, it destroys the constructive character of the intensional system, since it makes type-checking undecidable [Hofmann 1997]. For this reason, it is not assumed in the most recent formulations of Martin-Löf type theories [Nordstrom et al. 2000] or in automated proof assistants like Coq [Bertot and Castéran 2004].

In intensional type theories, inductive types cannot be characterized by standard category-theoretic universal properties. For instance, in this setting it is not possible to show that there exists a definitionally-unique function out of the empty type with rules as in [Nordstrom et al. 2000, Section 5.2], thus making it impossible to prove that the empty type provides an initial object. Another consequence of this fact is that, if we attempt to define the type of natural numbers as a W-type in the usual way, then the usual elimination and computation rules for it are no longer derivable [Dybjer 1997]. Similarly, it is not possible to show the uniqueness of recursively-defined functions out of W-types. When interpreted categorically, the uniqueness of such functions translates into the initiality property of the associated polynomial functor algebra, which is why the correspondence between W-types and initial algebras fails in the intensional setting.

Due to this sort of poor behaviour of W-types, and other constructions, in the purely intensional setting, that system is often augmented by other extensionality principles that are somewhat weaker than the Reflection rule, such as Streicher's K-rule or the Uniqueness of Identity Proofs (UIP) [Streicher 1993], which has recently been reconsidered in the context of Observational Type Theory [Altenkirch et al. 2007]. Inductive types in such intermediate systems are somewhat better behaved, but still exhibit some undesirable properties, making them less useful for practical purposes than one might wish [McBride 2010]. Moreover, these intermediate systems seem to lack a clear conceptual basis: they neither intend to formalize constructive sets (like the extensional theory) nor is there a principled reason to choose these particular extensionality rules, beyond their practical advantages.

2.2. The system $\mathcal{H}$

We here take a different approach to inductive types in the intensional setting, namely, one motivated by the homotopical interpretation. It involves working over a dependent type theory $\mathcal{H}$ which has the following deduction rules on top of the standard structural rules:

— rules for identity types as stated above;
— rules for Σ-types as in [Nordstrom et al. 2000, Section 5.8];
— rules for Π-types as in [Garner 2009, Section 3.2];
— the Function Extensionality axiom (FE), *i.e.* the axiom asserting that for every $f, g : A \to B$, the type

$$(\Pi x : A)\mathsf{Id}_B(\mathsf{ap}(f,x), \mathsf{ap}(g,x)) \to \mathsf{Id}_{A\to B}(f,g)$$

is inhabited.

Here, we have used the notation $A \to B$ to indicate function types, defined via Π-types in the usual way. Similarly, we will write $A \times B$ to denote the binary product of two types as usually defined via Σ-types.

Remarks

(i) The rules for Π-types of $\mathcal{H}$ are derivable from those in [Nordstrom et al. 2000, Section 5.4]. For simplicity, we will write $f(a)$ or fa instead of $\mathsf{ap}(f,a)$.
(ii) As shown in [Voevodsky 2010a], the principle of propositional function extensionality stated above implies the corresponding principle for dependent functions, *i.e.*

$$(\Pi x : A)\mathsf{Id}_{B(x)}(fx, gx) \to \mathsf{Id}_{(\Pi x : A)B(x)}(f,g)\,.$$

(iii) The following form of the η-rule for Σ-types is derivable:

$$\frac{c : (\Sigma x : A)B(x)}{\eta_\Sigma(c) : \mathsf{Id}(c, \mathsf{pair}(\pi_1 c\,, \pi_2 c))}\,,$$

where π_1 and π_2 are the projections. This can be proved by Σ-elimination, without FE.
(iv) The following form of the η-rule for Π-types is derivable:

$$\frac{f : (\Pi x : A)B(x)}{\eta_\Pi(f) : \mathsf{Id}(f,\ \lambda x.\, fx)}$$

This is an immediate consequence of FE and clearly implies the corresponding η-rule for function types.
(v) $\mathcal{H}$ does *not* include the η-rules as definitional equalities, either for Σ-types or for Π-types (as is done in [Goguen and Luo 1993]).

(vi) The type theory $\mathcal{H}$ will serve as the background theory for our study of inductive types and W-types. For this reason, we need not assume it to have any primitive types.

This particular combination of rules is motivated by the fact that $\mathcal{H}$ has a clear homotopy-theoretic sematics. Indeed, the type theory $\mathcal{H}$ is a subsystem of the type theory used in Voevodsky's Univalent Foundations library [Voevodsky 2010a]. In particular, the Function Extensionality axiom is formally implied by Voevodsky's Univalence axiom [Voevodsky 2009], which is also valid in homotopy-theoretic models, but will not be needed here. Note that, while the Function Extensionality axiom is valid also in set-theoretic models, the Univalence axiom is not. Although $\mathcal{H}$ has a straightforward set-theoretical semantics, we stress that it does not have any global extensionality rules, like the identity reflection rule, K, or UIP. This makes it also compatible with "higher-dimensional" interpretations such as the groupoid model [Hofmann and Streicher 1998], in which the rules of $\mathcal{H}$ are also valid.

2.3. Homotopical semantics

The homotopical semantics of $\mathcal{H}$ is based on the idea that an identity term $p : \mathsf{Id}_A(a, b)$ is (interpreted as) a path $p : a \rightsquigarrow b$ between the points a and b in the space A. More generally, the interpretations of terms $a(x)$ and $b(x)$ with free variables will be continuous functions into the space A, and an identity term $p(x) : \mathsf{Id}_A\big(a(x), b(x)\big)$ is then a continuous family of paths, *i.e.* a homotopy between the continuous functions. Now, the main import of the Id-elimination rule is that type dependency must respect identity, in the following sense: given a dependent type

$$x : A \vdash B(x) : \mathsf{type}\,, \tag{2}$$

and $p : \mathsf{Id}_A(a, b)$, there is then a *transport* function

$$p_! : B(a) \to B(b),$$

which is defined by Id-elimination, taking for $x : A$ the function $\mathsf{refl}(x)_! : B(x) \to B(x)$ to be the identity on $B(x)$. Semantically, given that an identity term $p : \mathsf{Id}_A(a, b)$ is interpreted as a path $p : a \rightsquigarrow b$, this means that a dependent type as in (2) must be interpreted as a space $B \to A$, fibered over the space A, and that the judgement

$$x, y : A \vdash \mathsf{Id}_A(x, y) : \mathsf{type}$$

is interpreted as the canonical fibration $A^I \to A \times A$ of the path space A^I over $A \times A$. For a more detailed overview of the homotopical interpretation, see [Awodey 2010].

Independently of this interpretation, each type A can be shown to carry the structure of a weak ω-groupoid in the sense of [Batanin 1998; Leinster 2004] with the elements of A as objects, identity proofs $p : \mathsf{Id}_A(a, b)$ as morphisms and elements of iterated identity types as n-cells [van den Berg and Garner 2011; Lumsdaine 2010b]. Furthermore, $\mathcal{H}$ determines a weak ω-category $\mathcal{C}(\mathcal{H})$ having types as 0-cells, elements $f : A \to B$ as 1-cells, and elements of (iterated) identity types as n-cells [Lumsdaine 2010a]. The relation between the weak ω-category structure of $\mathcal{C}(\mathcal{H})$ and the homotopical interpretation of intensional type theories closely mirrors that between higher category theory and homotopy theory in modern algebraic topology, and some methods developed in the latter setting are also applicable in type theory. For instance, the topological notion of contractibility admits the following type-theoretic counterpart, originally introduced by Voevodsky in [Voevodsky 2010a].

Definition 2.1. A type A is called *contractible* if the type

$$\mathsf{is\text{-}contr}(A) \coloneqq (\Sigma x : A)(\Pi y : A)\mathsf{Id}_A(x, y) \tag{3}$$

is inhabited.

The type $\mathsf{is\text{-}contr}(A)$ can be seen as the propositions-as-types translation of the formula stating that A has a unique element. However, its homotopical interpretation is as a space that is inhabited if and only if the space interpreting A is contractible in the usual topological sense. The notion of contractibility can be used to articulate the world of types into different homotopical dimensions, or *h-levels* [Voevodsky 2010a]. This classification has proven to be quite useful in understanding intensional type theory. For example, it permits the definition of new notions of *proposition* and *set* which provide a useful alternative to the standard approach to formalization of mathematics in type theory [Voevodsky 2010a].

Remark 2.2. If A is a contractible type, then for every $a, b : A$, the type $\mathsf{Id}_A(a, b)$ is again contractible. This can be proved by Id-elimination [Awodey et al. 2011].

Let us also recall from [Voevodsky 2010a] the notions of weak equivalence and homotopy equivalence. To do this, we need to fix some notation. For $f : A \to B$ and $y : B$, define the type

$$\mathsf{hfiber}(f, y) \coloneqq (\Sigma x : A)\mathsf{Id}_B(fx, y)\,.$$

We refer to this type as the *homotopy fiber* of f at y.

Definition 2.3. Let $f : A \to B$.

— We say that f is a *weak equivalence* if the type

$$\mathsf{isweq(f)} \coloneqq (\Pi y : B)\,\mathsf{is\text{-}contr}(\mathsf{hfiber}(f, y))$$

is inhabited.

— We say that f is a *homotopy equivalence* if there exist a function $g : B \to A$ and elements

$$\eta : (\Pi x : A)\mathsf{Id}(gfx, x)\,,$$
$$\varepsilon : (\Pi y : B)\mathsf{Id}(fgy, y)\,.$$

It is an *adjoint homotopy equivalence* if there are also terms

$$p : (\Pi x : A)\mathsf{Id}(\varepsilon_{fx}\,, f\,\eta_x)\,,$$
$$q : (\Pi y : B)\mathsf{Id}(\eta_{gy}\,, g\,\varepsilon_y)\,,$$

where the same notation for both function application and the action of a function on an identity proof (which is easily definable by Id-elimination), and we write α_x instead of $\alpha(x)$ for better readability.

The type $\mathsf{isweq(f)}$ can be seen as the propositions-as-types translation of the formula asserting that f is bijective, while homotopy equivalence is evidently a form of isomorphism. Thus it is a pleasant fact that a function is a weak equivalence if and only if it is a homotopy equivalence [Voevodsky 2010a]. We also note that all type-theoretic constructions are homotopy invariant, in the sense that they respect this relation of equivalence, a fact which is exploited by the Univalence axiom [Voevodsky 2009].

In Section 5.3 below, these and related homotopy-theoretic insights will be used to study inductive types, but first we must briefly review some basic facts about inductive types in the extensional setting.

3. INDUCTIVE TYPES IN EXTENSIONAL TYPE THEORY

We briefly recall here the theory of inductive types, particularly W-types, in fully extensional type theories. As an especially simple running example, we first consider the type 2 of Boolean truth values [Nordstrom et al. 2000, Section 5.1].

3.1. The type $\mathsf{2}$

The type $\mathsf{2}$ is not a W-type, but can be formulated as an inductive type in the familiar way by means of formation, introduction, elimination, and computation rules:

— $\mathsf{2}$-formation rule.

$$\mathsf{2} : \mathcal{U}_i \,.$$

— $\mathsf{2}$-introduction rules.

$$0 : \mathsf{2}\,, \qquad 1 : \mathsf{2}\,.$$

— $\mathsf{2}$-elimination rule ("induction").

$$\frac{x : \mathsf{2} \vdash E(x) : \mathcal{U}_i \qquad e_0 : E(0) \qquad e_1 : E(1) \qquad b : \mathsf{2}}{\mathsf{2ind}_{\mathcal{U}_i}(x.E, e_0, e_1, b) : E(b)}$$

— $\mathsf{2}$-computation rules.

$$\frac{x : \mathsf{2} \vdash E(x) : \mathcal{U}_i \qquad e_0 : E(0) \qquad e_1 : E(1)}{\begin{cases} \mathsf{2ind}_{\mathcal{U}_i}(x.E, e_0, e_1, 0) \equiv e_0 : E(0)\,, \\ \mathsf{2ind}_{\mathcal{U}_i}(x.E, e_0, e_1, 1) \equiv e_1 : E(1)\,. \end{cases}}$$

The above rules imply a non-dependent (simple) version of elimination and the corresponding computation rules:

— Simple $\mathsf{2}$-elimination rule ("recursion").

$$\frac{C : \mathcal{U}_i \qquad c_0 : C \qquad c_1 : C \qquad b : \mathsf{2}}{\mathsf{2rec}_{\mathcal{U}_i}(C, c_0, c_1, b) : C}$$

— Simple $\mathsf{2}$-computation rules.

$$\frac{C : \mathcal{U}_i \qquad c_0 : C \qquad c_1 : C}{\begin{cases} \mathsf{2rec}_{\mathcal{U}_i}(C, c_0, c_1, 0) \equiv c_0 : C\,, \\ \mathsf{2rec}_{\mathcal{U}_i}(C, c_0, c_1, 1) \equiv c_1 : C\,. \end{cases}}$$

Furthermore, the induction principle implies the following uniqueness principles, which state that any two functions out of $\mathsf{2}$ which agree on the constructors are (pointwise) equal:

— $\mathsf{2}$-uniqueness principle.

$$\frac{\begin{array}{ccc} b : \mathsf{2} \quad x : \mathsf{2} \vdash E(x) : \mathcal{U}_i & e_0 : E(0) & e_1 : E(1) \\ x : \mathsf{2} \vdash f(x) : E(x) & f(0) \equiv e_0 & f(1) \equiv e_1 \\ x : \mathsf{2} \vdash g(x) : E(x) & g(0) \equiv e_0 & g(1) \equiv e_1 \end{array}}{f(b) \equiv g(b) : E(b)}$$

— Simple $\mathsf{2}$-uniqueness principle.

$$\frac{\begin{array}{ccc} b : \mathsf{2} \quad C : \mathcal{U}_i & c_0 : C & c_1 : C \\ x : \mathsf{2} \vdash f(x) : C & f(0) \equiv c_0 & f(1) \equiv c_1 \\ x : \mathsf{2} \vdash g(x) : C & g(0) \equiv c_0 & g(1) \equiv c_1 \end{array}}{f(b) \equiv g(b) : C}$$

Remark: In this particular case, we could also write $f(0) \equiv g(0) : E(0)$ and $f(1) \equiv g(1) : E(1)$ in the premises, avoiding the mention of e_0 and e_1 altogether (and similarly for the simple uniqueness principle). However, the form we chose follows a general schema (see Defs. 3.4 and 3.8) which scales to more complicated inductive types.

We can prove the dependent uniqueness principle by using the induction principle with the type family $x \mapsto f(x) =_{E(x)} g(x)$; the cases when $x :\equiv 0$ and $x :\equiv 1$ are straightforward. This gives us pointwise propositional equality between f and g; we appeal to the identity reflection rule to finish the proof. The simple uniqueness principle clearly follows from the dependent one.

We now introduce some terminology in order to talk about the above rules more economically. The elimination and computation rules motivate the following definitions:

Definition 3.1. Define the type of 2-*algebras* on a universe $\mathcal{U}_i$ as

$$\mathsf{2\text{-}Alg}_{\mathcal{U}_i} :\equiv \Sigma_{C:\mathcal{U}_i} C \times C$$

Definition 3.2. Define the type of *fibered* 2-*algebras* on a universe $\mathcal{U}_j$ over $\mathcal{X}$: $\mathsf{2\text{-}Alg}_{\mathcal{U}_i}$ by

$$\mathsf{2\text{-}Fib\text{-}Alg}_{\mathcal{U}_j}\ (C, c_0, c_1) :\equiv \Sigma_{E:C\to\mathcal{U}_j} E(c_0) \times E(c_1)$$

Definition 3.3. Given algebras $\mathcal{X} : \mathsf{2\text{-}Alg}_{\mathcal{U}_i}$ and $\mathcal{Y} : \mathsf{2\text{-}Alg}_{\mathcal{U}_j}$, define the type of 2-*homomorphisms* from $\mathcal{X}$ to $\mathcal{Y}$ by

$$\mathsf{2\text{-}Hom}\ (C, c_0, c_1)\ (D, d_0, d_1) :\equiv \Sigma_{f:C\to D}(f(c_0) = d_0) \times (f(c_1) = d_1)$$

Definition 3.4. Given algebras $\mathcal{X} : \mathsf{2\text{-}Alg}_{\mathcal{U}_i}$ and $\mathcal{Y} : \mathsf{2\text{-}Fib\text{-}Alg}_{\mathcal{U}_j}\ \mathcal{X}$, define the type of *fibered* 2-*homomorphisms* from $\mathcal{X}$ to $\mathcal{Y}$ by

$$\mathsf{2\text{-}Fib\text{-}Hom}\ (C, c_0, c_1)\ (E, e_0, e_1) :\equiv \Sigma_{(f:\Pi_{x:C} E(x))}(f(c_0) = e_0) \times (f(c_1) = e_1)$$

We note that in order to form the type of homomorphisms, we had to use propositional rather than definitional equality. Of course, in the setting of an extensional type theory this distinction is immaterial; however, it will become important in Chpt. 5.3.

Definition 3.5. An algebra $\mathcal{X} : \mathsf{2\text{-}Alg}_{\mathcal{U}_i}$ *satisfies the recursion principle* on a universe $\mathcal{U}_j$ if for any algebra $\mathcal{Y} : \mathsf{2\text{-}Alg}_{\mathcal{U}_j}$ there exists a 2-homomorphism between $\mathcal{X}$ and $\mathcal{Y}$:

$$\mathsf{has\text{-}2\text{-}rec}_{\mathcal{U}_j}(\mathcal{X}) :\equiv \Pi_{(\mathcal{Y}:\mathsf{2\text{-}Alg}_{\mathcal{U}_j})} \mathsf{2\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}$$

Definition 3.6. An algebra $\mathcal{X} : \mathsf{2\text{-}Alg}_{\mathcal{U}_i}$ *satisfies the induction principle* on a universe $\mathcal{U}_j$ if for any fibered algebra $\mathcal{Y} : \mathsf{2\text{-}Fib\text{-}Alg}_{\mathcal{U}_j}\ \mathcal{X}$ there exists a fibered 2-homomorphism between $\mathcal{X}$ and $\mathcal{Y}$:

$$\mathsf{has\text{-}2\text{-}ind}_{\mathcal{U}_j}(\mathcal{X}) :\equiv \Pi_{(\mathcal{Y}:\mathsf{2\text{-}Fib\text{-}Alg}_{\mathcal{U}_j}\ \mathcal{X})} \mathsf{2\text{-}Fib\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}$$

The uniqueness principles motivate the following definitions:

Definition 3.7. An algebra $\mathcal{X} : \mathsf{2\text{-}Alg}_{\mathcal{U}_i}$ satisfies the *recursion uniqueness principle* on a universe $\mathcal{U}_j$ if for any algebra $\mathcal{Y} : \mathsf{2\text{-}Alg}_{\mathcal{U}_j}$ any two 2-homomorphisms between $\mathcal{X}$ and $\mathcal{Y}$ are equal:

$$\mathsf{has\text{-}2\text{-}rec\text{-}uniq}_{\mathcal{U}_j}(\mathcal{X}) :\equiv \Pi_{(\mathcal{Y}:\mathsf{2\text{-}Alg}_{\mathcal{U}_j})} \mathsf{is\text{-}prop}(\mathsf{2\text{-}Hom}\ \mathcal{X}\ \mathcal{Y})$$

Definition 3.8. An algebra $\mathcal{X} : \mathsf{2\text{-}Alg}_{\mathcal{U}_i}$ satisfies the *induction uniqueness principle* on a universe $\mathcal{U}_j$ if for any fibered algebra $\mathcal{Y} : \mathsf{2\text{-}Fib\text{-}Alg}_{\mathcal{U}_j}\ \mathcal{X}$ any two fibered 2-homomorphisms between $\mathcal{X}$ and $\mathcal{Y}$ are equal:

$$\mathsf{has\text{-}2\text{-}ind\text{-}uniq}_{\mathcal{U}_j}(\mathcal{X}) :\equiv \Pi_{(\mathcal{Y}:\mathsf{2\text{-}Fib\text{-}Alg}_{\mathcal{U}_j}\ \mathcal{X})} \mathsf{is\text{-}prop}(\mathsf{2\text{-}Fib\text{-}Hom}\ \mathcal{X}\ \mathcal{Y})$$

The uniqueness principles as in definitions 3.7, 3.8 require that any two homomorphisms $\mu :\equiv (f, \gamma_0, \gamma_1)$ and $\nu :\equiv (g, \delta_0, \delta_1)$ be equal as *tuples*; however, in the presence of the UIP axiom this is equivalent to saying that their first components agree, i.e., $f = g$ (and hence $f \equiv g$).

Definition 3.9. An algebra $\mathcal{X} : \textsf{2-Alg}_{\mathcal{U}_i}$ is *initial* on a universe $\mathcal{U}_j$ if for any algebra $\mathcal{Y} : \textsf{2-Alg}_{\mathcal{U}_j}$ there exists a unique 2-homomorphism between $\mathcal{X}$ and $\mathcal{Y}$:

$$\textsf{is-2-init}_{\mathcal{U}_j}(\mathcal{X}) :\equiv \Pi_{(\mathcal{Y}:\textsf{2-Alg}_{\mathcal{U}_j})}\textsf{is-contr}(\textsf{2-Hom}\ \mathcal{X}\ \mathcal{Y})$$

The contractibility requirement precisely captures the notion of initiality: in the presence of the identity reflection rule, a contractible type is one which contains a definitionally unique element.

The following lemma in particular implies that it is not necessary to have a "fibered" version of the initiality property, which quantifies over all fibered algebras $\mathcal{Y} : \textsf{2-Fib-Alg}_{\mathcal{U}_j}\ \mathcal{X}$.

LEMMA 3.10. *In $\mathcal{H}_{\text{ext}}$, if an algebra $\mathcal{X} : \textsf{2-Alg}_{\mathcal{U}_i}$ satisfies the induction principle on the universe $\mathcal{U}_j$, it also satisfies the induction uniqueness principle on $\mathcal{U}_j$. In other words, we have*

$$\textsf{has-2-ind}_{\mathcal{U}_j}(\mathcal{X}) \quad \to \quad \textsf{has-2-ind-uniq}_{\mathcal{U}_j}(\mathcal{X})$$

PROOF. Let an algebra $(C, c_0, c_1) : \textsf{2-Alg}_{\mathcal{U}_i}$ be given. To prove the induction uniqueness principle, take any algebra $(E, e_0, e_1) : \textsf{2-Fib-Alg}_{\mathcal{U}_j}\ (C, c_0, c_1)$ and homomorphisms $(f, \gamma_0, \gamma_1), (g, \delta_0, \delta_1) : \textsf{2-Fib-Hom}\ (C, c_0, c_1)\ (E, e_0, e_1)$. Because of UIP, showing $(f, \gamma_0, \gamma_1) = (g, \delta_0, \delta_1)$ is equivalent to showing $f = g$. For the latter, we use the induction principle with the fibered algebra $\big(x \mapsto f(x) = g(x), \textsf{refl}(f(c_0)), \textsf{refl}(g(c_0))\big)$. This is indeed well-typed since γ_0, δ_0 give us $f(c_0) = e_0 = g(c_0)$ and γ_1, δ_1 give us $f(c_1) = e_1 = g(c_1)$.

The first component of the resulting homomorphism together with the function extensionality principle then give us $f = g$. $\square$

COROLLARY 3.11. *In $\mathcal{H}_{\text{ext}}$, if an algebra $\mathcal{X} : \textsf{2-Alg}_{\mathcal{U}_i}$ satisfies the induction principle on the universe $\mathcal{U}_j$, it is initial on $\mathcal{U}_j$. In other words, we have*

$$\textsf{has-2-ind}_{\mathcal{U}_j}(\mathcal{X}) \quad \to \quad \textsf{is-2-init}_{\mathcal{U}_j}(\mathcal{X})$$

LEMMA 3.12. *In $\mathcal{H}_{\text{ext}}$, if an algebra $\mathcal{X} : \textsf{2-Alg}_{\mathcal{U}_i}$ satisfies the recursion and recursion uniqueness principles on the universe $\mathcal{U}_j$ and $j \geq i$, then it satisfies the induction principle on $\mathcal{U}_j$. In other words, we have*

$$\textsf{has-2-rec}_{\mathcal{U}_j}(\mathcal{X}) \times \textsf{has-2-rec-uniq}_{\mathcal{U}_j}(\mathcal{X}) \quad \to \quad \textsf{has-2-ind}_{\mathcal{U}_j}(\mathcal{X})$$

provided $j \geq i$.

PROOF. Let algebras $(C, c_0, c_1) : \textsf{2-Alg}_{\mathcal{U}_i}$ and $(E, e_0, e_1) : \textsf{2-Fib-Alg}_{\mathcal{U}_j}\ (C, c_0, c_1)$ be given. We use the recursion principle with the algebra $\big(\Sigma_{x:C} E(x), (c_0, e_0), (c_1, e_1)\big)$. We note that the carrier type belongs to $\mathcal{U}_j$ as $i \leq j$. This gives us a homomorphism (u, θ_0, θ_1), where $u : C \to \Sigma_{x:C} E(x)$, $\theta_0 : u(c_0) = (c_0, e_0)$, and $\theta_1 : u(c_1) = (c_1, e_1)$. We can now form two homomorphisms $\big(\textsf{fst} \circ u, \textsf{refl}(c_0), \textsf{refl}(c_1)\big), \big(\textsf{id}_C, \textsf{refl}(c_0), \textsf{refl}(c_1)\big) : \textsf{2-Hom}\ (C, c_0, c_1)\ (C, c_0, c_1)$; the former type-checks due to θ_0, θ_1. The recursion uniqueness principle tells us that these homomorphisms are equal. Thus, we have $\textsf{fst} \circ u = \textsf{id}_C$ and in particular $\textsf{fst} \circ u \equiv \textsf{id}_C$.

We can thus define the desired fibered homomorphism as $\big(\textsf{snd} \circ u, \textsf{refl}(e_0), \textsf{refl}(e_1)\big) : \textsf{2-Fib-Hom}\ (C, c_0, c_1)\ (E, e_0, e_1)$. $\square$

COROLLARY 3.13. *In $\mathcal{H}_{\text{ext}}$, the following conditions on an algebra $\mathcal{X} : \mathsf{2\text{-}Alg}_{\mathcal{U}_i}$ are equivalent:*

(1) *$\mathcal{X}$ satisfies the induction principle on the universe $\mathcal{U}_j$*
(2) *$\mathcal{X}$ satisfies the recursion and recursion uniqueness principles on the universe $\mathcal{U}_j$*
(3) *$\mathcal{X}$ is initial on the universe $\mathcal{U}_j$*

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

$$\mathsf{has\text{-}2\text{-}ind}_{\mathcal{U}_j}(\mathcal{X}) \;\simeq\; \mathsf{has\text{-}2\text{-}rec}_{\mathcal{U}_j}(\mathcal{X}) \times \mathsf{has\text{-}2\text{-}rec\text{-}uniq}_{\mathcal{U}_j}(\mathcal{X}) \;\simeq\; \mathsf{is\text{-}2\text{-}init}_{\mathcal{U}_j}(\mathcal{X})$$

provided $j \geq i$. Furthermore, all 3 conditions are mere propositions.

PROOF. Conditions (1) and (3) are mere propositions by Lem. **??**[1] (for the latter), EdN:1
Lem. **??**[2], 3.10 (for the former), and the fact that a family of mere propositions is itself EdN:2
a mere proposition. □

We can thus characterize the type 2 using the universal property of initiality as follows.

COROLLARY 3.14. *In $\mathcal{H}_{\text{ext}}$ extended with the type 2, the algebra $(2, 0, 1) : \mathsf{2\text{-}Alg}_{\mathcal{U}_0}$ is initial on any universe $\mathcal{U}_j$.*

COROLLARY 3.15. *In $\mathcal{H}_{\text{ext}}$ extended with an algebra $\mathcal{X} : \mathsf{2\text{-}Alg}_{\mathcal{U}_0}$ which is initial on any universe $\mathcal{U}_j$, the type 2 is definable.*

PROOF. We have an algebra $\cdot \vdash \mathcal{X} : \mathsf{2\text{-}Alg}_{\mathcal{U}_0}$ such that for any j, there exists a term $\cdot \vdash h_j : \mathsf{is\text{-}2\text{-}init}_{\mathcal{U}_j}(\mathcal{X})$. Since the requirement $j \geq 0$ always holds, Cor. 3.13 implies that for any j, we have a term $\cdot \vdash r_j : \mathsf{has\text{-}2\text{-}ind}_{\mathcal{U}_j}(\mathcal{X})$. This implies that the type 2 is definable. □

3.2. W-types

We recall the rules for W-types from [Martin-Löf 1984]; to state them more conveniently, we sometimes write W instead of $\mathsf{W}^{\mathcal{U}_i}_{x:A}B(x)$; $\mathsf{sup}(a,t)$ instead of $\mathsf{sup}^{A,x.B}_{\mathcal{U}_i}(a,t)$; and $\mathsf{wind}(m)$ instead of $\mathsf{wind}^{A,x.B}_{\mathcal{U}_j}\big(w.E, a.t.s.e, m\big)$.

— W-formation rule.

$$\frac{A : \mathcal{U}_i \qquad x : A \vdash B(x) : \mathcal{U}_i}{\mathsf{W}^{\mathcal{U}_i}_{x:A}B(x)}$$

— W-introduction rule.

$$\frac{A : \mathcal{U}_i \qquad x : A \vdash B(x) : \mathcal{U}_i \qquad a : A \qquad t : B(a) \to \mathsf{W}}{\mathsf{sup}^{A,x.B}_{\mathcal{U}_i}(a,t) : \mathsf{W}}$$

— W-elimination rule.

$$\frac{\begin{array}{c} A : \mathcal{U}_i \qquad x : A \vdash B(x) : \mathcal{U}_i \qquad m : \mathsf{W} \\ w : \mathsf{W} \vdash E(w) : \mathcal{U}_j \qquad x : A, p : B(x) \to \mathsf{W}, r : \Pi_{b:B(x)} E(p\ b) \vdash e(x,p,r) : E(\mathsf{sup}(x,p)) \end{array}}{\mathsf{wind}^{A,x.B}_{\mathcal{U}_j}\big(w.E, x.p.r.e, m\big) : E(m)}$$

— W-computation rule.

[1] EDNOTE: Which says $\mathsf{is\text{-}prop}(\mathsf{is\text{-}contr}(A))$
[2] EDNOTE: Which says $\mathsf{is\text{-}prop}(A) \simeq A \to \mathsf{is\text{-}prop}(A)$

$$\frac{\begin{array}{c} A : \mathcal{U}_i \qquad x : A \vdash B(x) : \mathcal{U}_i \qquad a : A \qquad t : B(a) \to \mathsf{W} \\ w : \mathsf{W} \vdash E(w) : \mathcal{U}_j \qquad x : A, p : B(x) \to \mathsf{W}, r : \Pi_{b:B(x)} E(p\ b) \vdash e(x,p,r) : E(\mathsf{sup}(x,p)) \end{array}}{\mathsf{wind}(\mathsf{sup}(a,t)) \equiv e(a,t,\lambda_{b:B(a)} \mathsf{wind}(t\ b)) : E(\mathsf{sup}(a,t))}$$

W-types can be seen informally as the free algebras for signatures with operations of possibly infinite arity, but no equations. Indeed, the premises of the formation rule above can be thought of as specifying a signature that has the elements of A as operations and in which the arity of $a : A$ is the cardinality of the type $B(a)$. Then, the introduction rule specifies the canonical way of forming an element of the free algebra, and the elimination rule can be seen as the propositions-as-types translation of the appropriate induction principle.

As before, the above rules imply a non-dependent version of elimination and the corresponding computation rule, where we write $\mathsf{wrec}(m)$ instead of $\mathsf{wrec}^{A,x.B}_{\mathcal{U}_j}(C, x.r.c, m)$ where appropriate.

— Simple W-elimination rule.

$$\frac{A : \mathcal{U}_i \qquad x : A \vdash B(x) : \mathcal{U}_i \qquad m : \mathsf{W} \qquad C : \mathcal{U}_j \qquad x : A, r : B(x) \to C \vdash c(x,r) : C}{\mathsf{wrec}^{A,x.B}_{\mathcal{U}_j}(C, x.r.c, m) : C}$$

— Simple W-computation rule.

$$\frac{\begin{array}{c} A : \mathcal{U}_i \qquad x : A \vdash B(x) : \mathcal{U}_i \\ a : A \qquad t : B(a) \to \mathsf{W} \qquad C : \mathcal{U}_j \qquad x : A, r : B(x) \to C \vdash c(x,r) : C \end{array}}{\mathsf{wrec}(\mathsf{sup}(a,t)) \equiv c(a, \lambda_{b:B(a)} \mathsf{wrec}(t\ b)) : C}$$

The induction principle also implies the following uniqueness principles, which state that any two functions out of W which satisfy the same recurrence are pointwise equal:

— W-uniqueness principle.

$$\frac{\begin{array}{c} A : \mathcal{U}_i \qquad x : A \vdash B(x) : \mathcal{U}_i \qquad m : \mathsf{W} \\ w : \mathsf{W} \vdash E(w) : \mathcal{U}_j \qquad x : A, p : B(x) \to \mathsf{W}, r : \Pi_{b:B(x)} E(p\ b) \vdash e(x,p,r) : E(\mathsf{sup}(x,p)) \\ w : \mathsf{W} \vdash f(w) : E(w) \qquad x : A, p : B(x) \to \mathsf{W} \vdash f(\mathsf{sup}(x,p)) \equiv e(x,p,\lambda_{b:B(x)} f(p\ b)) \\ w : \mathsf{W} \vdash g(w) : E(w) \qquad x : A, p : B(x) \to \mathsf{W} \vdash g(\mathsf{sup}(x,p)) \equiv e(x,p,\lambda_{b:B(x)} g(p\ b)) \end{array}}{f(m) \equiv g(m) : E(m)}$$

— Simple W-uniqueness principle.

$$\frac{\begin{array}{c} A : \mathcal{U}_i \qquad x : A \vdash B(x) : \mathcal{U}_i \qquad m : \mathsf{W} \qquad C : \mathcal{U}_j \qquad x : A, r : B(x) \to C \vdash c(x,r) : C \\ w : \mathsf{W} \vdash f(w) : C \qquad x : A, p : B(x) \to \mathsf{W} \vdash f(\mathsf{sup}(x,p)) \equiv c(x, \lambda_{b:B(x)} f(p\ b)) \\ w : \mathsf{W} \vdash g(x) : C \qquad x : A, p : B(x) \to \mathsf{W} \vdash g(\mathsf{sup}(x,p)) \equiv c(x, \lambda_{b:B(x)} g(p\ b)) \end{array}}{f(m) \equiv g(m) : C}$$

As before, we can prove the dependent uniqueness principle by using the induction principle with the type family $w \mapsto f(w) =_{E(w)} g(w)$. For this we need to show that for any x, p we have $f(\mathsf{sup}(x,p)) = g(\mathsf{sup}(x,p))$ under the induction hypothesis $\Pi_{b:B(x)} f(p\ b) = g(p\ b)$. Using this hypothesis with function extensionality we get $\lambda_{b:B(x)} f(p\ b) = \lambda_{b:B(x)} g(p\ b)$. Together with the premises of the uniqueness rule this gives us $f(\mathsf{sup}(a,t)) = g(\mathsf{sup}(a,t))$ as desired. The induction principle thus gives us pointwise propositional equality between f and g and we appeal to the identity reflection rule to finish the proof. The simple uniqueness principle again follows from the dependent one.

We can now formulate the notions of algebras, homomorphisms, etc. accordingly:

Definition 3.16. For $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, define the type of W-*algebras* on a universe $\mathcal{U}_j$ as

$$\mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A,B) \;\coloneqq \Sigma_{C:\mathcal{U}_j}\Pi_{a:A}(B(a)\to C)\to C$$

Definition 3.17. For $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, define the type of *fibered* W-*algebras* on a universe $\mathcal{U}_k$ over $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A,B)$ by

$$\mathsf{W\text{-}Fib\text{-}Alg}_{\mathcal{U}_k}(A,B)\;(C,c) \coloneqq \Sigma_{E:C\to\mathcal{U}_k}\Pi_{a:A}\Pi_{t:B(a)\to C}\big(\Pi_{b:B(a)}E(t\;b)\big)\to E(c(a,t))$$

Definition 3.18. For $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A,B)$, $\mathcal{Y} : \mathsf{W\text{-}Alg}_{\mathcal{U}_k}(A,B)$, define the type of W-*homomorphisms* from $\mathcal{X}$ to $\mathcal{Y}$ by

$$\mathsf{W\text{-}Hom}\;(C,c)\;(D,d) \coloneqq \Sigma_{f:C\to D}\Pi_{a:A}\Pi_{t:B(a)\to C}f(c(a,t)) = d(a,\lambda_{b:B(a)}f(t\;b))$$

Definition 3.19. For $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A,B)$, $\mathcal{Y} : \mathsf{W\text{-}Fib\text{-}Alg}_{\mathcal{U}_k}(A,B)\;\mathcal{X}$, define the type of *fibered* W-*homomorphisms* from $\mathcal{X}$ to $\mathcal{Y}$ by

$$\mathsf{W\text{-}Fib\text{-}Hom}\;(C,c)\;(E,e) \coloneqq \Sigma_{(f:\Pi_{x:C}E(x))}\Pi_{a:A}\Pi_{t:B(a)\to C}f(c(a,t)) = e(a,t,\lambda_{b:B(a)}f(t\;b))$$

Definition 3.20. For $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, an algebra $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A,B)$ *satisfies the recursion principle* on a universe $\mathcal{U}_k$ if for any algebra $\mathcal{Y} : \mathsf{W\text{-}Alg}_{\mathcal{U}_k}(A,B)$ there exists a W-homomorphism between $\mathcal{X}$ and $\mathcal{Y}$:

$$\mathsf{has\text{-}W\text{-}rec}_{\mathcal{U}_k}(\mathcal{X}) \coloneqq \Pi_{(\mathcal{Y}:\mathsf{W\text{-}Alg}_{\mathcal{U}_k}(A,B))}\mathsf{W\text{-}Hom}\;\mathcal{X}\;\mathcal{Y}$$

Definition 3.21. For $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, an algebra $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A,B)$ *satisfies the induction principle* on a universe $\mathcal{U}_k$ if for any fibered algebra $\mathcal{Y} : \mathsf{W\text{-}Fib\text{-}Alg}_{\mathcal{U}_k}(A,B)\;\mathcal{X}$ there exists a fibered W-homomorphism between $\mathcal{X}$ and $\mathcal{Y}$:

$$\mathsf{has\text{-}W\text{-}ind}_{\mathcal{U}_k}(\mathcal{X}) \coloneqq \Pi_{(\mathcal{Y}:\mathsf{W\text{-}Fib\text{-}Alg}_{\mathcal{U}_k}(A,B)\;\mathcal{X})}\mathsf{W\text{-}Fib\text{-}Hom}\;\mathcal{X}\;\mathcal{Y}$$

Definition 3.22. For $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, an algebra $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A,B)$ *satisfies the recursion uniqueness principle* on a universe $\mathcal{U}_k$ if for any algebra $\mathcal{Y} : \mathsf{W\text{-}Alg}_{\mathcal{U}_k}(A,B)$ any two W-homomorphisms between $\mathcal{X}$ and $\mathcal{Y}$ are equal:

$$\mathsf{has\text{-}W\text{-}rec\text{-}uniq}_{\mathcal{U}_k}(\mathcal{X}) \coloneqq \Pi_{(\mathcal{Y}:\mathsf{W\text{-}Alg}_{\mathcal{U}_k}(A,B))}\mathsf{is\text{-}prop}(\mathsf{W\text{-}Hom}\;\mathcal{X}\;\mathcal{Y})$$

Definition 3.23. For $A : \mathcal{U}_i$ and $B : A \to \mathcal{U}_i$, an algebra $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A,B)$ *satisfies the induction uniqueness principle* on a universe $\mathcal{U}_k$ if for any fibered algebra $\mathcal{Y} : \mathsf{W\text{-}Fib\text{-}Alg}_{\mathcal{U}_k}(A,B)\;\mathcal{X}$ any two fibered W-homomorphisms between $\mathcal{X}$ and $\mathcal{Y}$ are equal:

$$\mathsf{has\text{-}W\text{-}ind\text{-}uniq}_{\mathcal{U}_k}(\mathcal{X}) \coloneqq \Pi_{(\mathcal{Y}:\mathsf{W\text{-}Fib\text{-}Alg}_{\mathcal{U}_k}(A,B)\;\mathcal{X})}\mathsf{is\text{-}prop}(\mathsf{W\text{-}Fib\text{-}Hom}\;\mathcal{X}\;\mathcal{Y})$$

Definition 3.24. For $A : \mathcal{U}_i$ and $B : A \to \mathcal{U}_i$, an algebra $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A,B)$ is *initial* on a universe $\mathcal{U}_k$ if for any algebra $\mathcal{Y} : \mathsf{W\text{-}Alg}_{\mathcal{U}_k}(A,B)$ there exists a unique W-homomorphism between $\mathcal{X}$ and $\mathcal{Y}$:

$$\mathsf{is\text{-}W\text{-}init}_{\mathcal{U}_k}(\mathcal{X}) \coloneqq \Pi_{(\mathcal{Y}:\mathsf{W\text{-}Alg}_{\mathcal{U}_k}(A,B))}\mathsf{is\text{-}contr}(\mathsf{W\text{-}Hom}\;\mathcal{X}\;\mathcal{Y})$$

As before, the induction principle implies induction uniqueness:

LEMMA 3.25. *In $\mathcal{H}_{\mathrm{ext}}$, for $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, if an algebra $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A,B)$ satisfies the induction principle on the universe $\mathcal{U}_k$, it also satisfies the induction uniqueness principle on $\mathcal{U}_k$. In other words, we have*

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

PROOF. Let an algebra (C, c) : $\mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_j}(A, B)$ be given. To prove the induction uniqueness principle, take any algebra (E, e) : $\mathsf{W}\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_k}\ (C, c)$ and homomorphisms $(f, \gamma), (g, \delta)$: $\mathsf{W}\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\ (C, c)\ (E, e)$. Because of UIP, showing $(f, \gamma) = (g, \delta)$ is equivalent to showing $f = g$. For the latter, we use the induction principle with the fibered algebra $\big(x \mapsto f(x) = g(x); a, t, s \mapsto \mathsf{refl}(f(c(a, t)))\big)$. This is indeed well-typed since $\gamma(a, t)$ gives us $f(c(a, t)) = e(a, t, \lambda_{b:B(a)} f(t\ b))$; $\delta(a, t)$ gives us $g(c(a, t)) = e(a, t, \lambda_{b:B(a)} g(t\ b))$; and s together with function extensionality gives us $\lambda_{b:B(a)} f(t\ b) = \lambda_{b:B(a)} g(t\ b)$.

The first component of the resulting homomorphism together with the function extensionality principle then give us $f = g$. □

COROLLARY 3.26. *In $\mathcal{H}_{\mathrm{ext}}$, for $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, if an algebra $\mathcal{X}$: $\mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_j}(A, B)$ satisfies the induction principle on the universe $\mathcal{U}_k$, it is initial on $\mathcal{U}_k$. In other words, we have*

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

LEMMA 3.27. *In $\mathcal{H}_{\mathrm{ext}}$, for $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, if an algebra $\mathcal{X}$: $\mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_j}(A, B)$ satisfies the recursion and recursion uniqueness principles on the universe $\mathcal{U}_k$ and $k \geq j$, then it satisfies the induction principle on $\mathcal{U}_k$. In other words, we have*

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

provided $k \geq j$.

PROOF. Let algebras (C, c) : $\mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_j}(A, B)$ and (E, e) : $\mathsf{W}\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_k}\ (C, c)$ be given. We use the recursion principle with the algebra $(\Sigma_{x:C} E(x); a, s \mapsto d(a, s))$ where

$$d(a, s) :\equiv \Big(c\big(a, \lambda_{b:B(a)} \mathsf{fst}(s\ b)\big), e\big(a, \lambda_{b:B(a)} \mathsf{fst}(s\ b), \lambda_{b:B(a)} \mathsf{snd}(s\ b)\big)\Big)$$

We note that the carrier type belongs to $\mathcal{U}_k$ as $j \leq k$. This gives us a homomorphism (u, θ), where $u : C \to \Sigma_{x:C} E(x)$ and $\theta : \Pi_a \Pi_t u(c(a, t)) = d(a, \lambda_{b:B(a)} u(t\ b))$. We can now form two homomorphisms

$$\big(\mathsf{fst} \circ u; a, t \mapsto \mathsf{refl}_{c(a, \lambda_{b:B(a)} u(t\ b))}\big), \big(\mathsf{id}_C; a, t \mapsto \mathsf{refl}_{c(a,t)}\big) : \mathsf{W}\text{-}\mathsf{Hom}\ (C, c)\ (C, c)$$

The first one type-checks due to θ. The recursion uniqueness principle tells us that these homomorphisms are equal. Thus, $\mathsf{fst} \circ u = \mathsf{id}_C$ and in particular $\mathsf{fst} \circ u \equiv \mathsf{id}_C$.

We can thus define the desired fibered homomorphism as

$$\big(\mathsf{snd} \circ u; a, t \mapsto \mathsf{refl}(e(a, t, \lambda_{b:B(a)} \mathsf{snd}(u(t\ b))))\big) : \mathsf{W}\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\ (C, c)\ (E, e)$$

which type-checks due to θ and the fact that $\mathsf{fst} \circ u \equiv \mathsf{id}_C$. □

COROLLARY 3.28. *In $\mathcal{H}_{\mathrm{ext}}$, for $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, the following conditions on an algebra $\mathcal{X}$: $\mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_j}(A, B)$ are equivalent:*

(1) *$\mathcal{X}$ satisfies the induction principle on the universe $\mathcal{U}_k$*
(2) *$\mathcal{X}$ satisfies the recursion and recursion uniqueness principles on the universe $\mathcal{U}_k$*
(3) *$\mathcal{X}$ is initial on the universe $\mathcal{U}_k$*

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

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

provided $k \geq j$. Furthermore, all 3 conditions are mere propositions.

PROOF. Conditions (1) and (3) are mere propositions by Lem. **??** (for the latter), Lem. **??**, 3.10 (for the former), and the fact that a family of mere propositions is itself a mere proposition. □

We can thus characterize W-types using the universal property of initiality as follows.

COROLLARY 3.29. *In $\mathcal{H}_{\text{ext}}$ with* W*-types, for any $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, the algebra*

$$\Big(\mathsf{W}^{\mathcal{U}_i}_{x:A}B(x), \lambda_a\lambda_t\mathsf{sup}^{A,x.B(x)}_{\mathcal{U}_i}(a,t)\Big) : \mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_i}(A,B)$$

is initial on any universe $\mathcal{U}_j$.

COROLLARY 3.30. *In $\mathcal{H}_{\text{ext}}$ extended with an algebra $\mathcal{X}_{\mathcal{U}_i}(A,B) : \mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_i}(A,B)$ for any $\mathcal{U}_i$, $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, which is initial on any universe $\mathcal{U}_j$, Martin-Löf's* W*-types are definable.*

PROOF. For any i, the algebra $A : \mathcal{U}_i, B : A \to \mathcal{U}_i \vdash \mathcal{X}_{\mathcal{U}_i}(A,B) : \mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_i}(A,B)$ is such that for any j, there exists a term $A : \mathcal{U}_i, B : A \to \mathcal{U}_i \vdash h^i_j : \mathsf{is}\text{-}\mathsf{W}\text{-}\mathsf{init}_{\mathcal{U}_j}(\mathcal{X}_{\mathcal{U}_i}(A,B))$. By Cor. 3.13, for any $j \geq i$ there is a term $A : \mathcal{U}_i, B : A \to \mathcal{U}_i \vdash r^i_j : \mathsf{has}\text{-}\mathsf{W}\text{-}\mathsf{ind}_{\mathcal{U}_j}(\mathcal{X}_{\mathcal{U}_i}(A,B))$. Since universes are cumulative, this implies that such a term r^i_j exists *for any* j. This in turn implies that W-types are definable. $\square$

We conclude by the type-theoretic analogue of Lambek's lemma for W-types, which asserts that the structure map of an initial algebra is an equivalence. In $\mathcal{H}_{\text{ext}}$ any equivalence is also a type isomorphism, which then gives us the more familiar formulation of Lambek's lemma.

LEMMA 3.31. *Over $\mathcal{H}_{\text{ext}}$, for $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, if an algebra $(C,c) : \mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_j}(A,B)$ is initial on $\mathcal{U}_j$ and $j \geq i$, then the map from $\Sigma_{x:A}B(x) \to C$ to C given by c is an equivalence.*

PROOF. By abuse of notation we refer to both the curried and uncurried versions of the structure map by c. Since (C,c) is initial on $\mathcal{U}_j$, it satisfies the recursion principle on $\mathcal{U}_j$. We use it with the algebra

$$\Big(\Sigma_{x:A}B(x) \to C; a, s \mapsto \big(a, \lambda_{b:B(a)}c(s\ b)\big)\Big)$$

We note that the carrier type belongs to $\mathcal{U}_j$ as $i \leq j$. This gives us a homomorphism (u,θ) where $u : C \to \big(\Sigma_{x:A}B(x) \to C\big)$ and $\theta : \Pi_a\Pi_t u(c(a,t)) = \big(a, \lambda_{b:B(a)}c(u(t\ b))\big)$. We can now form two homomorphisms

$$\big(c \circ u; a, t \mapsto \mathsf{refl}_{c(a,\lambda_{b:B(a)}u(t\ b))}\big), \big(\mathsf{id}_C; a, t \mapsto \mathsf{refl}_{c(a,t)}\big) : \mathsf{W}\text{-}\mathsf{Hom}\ (C,c)\ (C,c)$$

where the first one type-checks due to θ. Since (C,c) is initial on $\mathcal{U}_j$, it satisfies the recursion uniqueness principle on $\mathcal{U}_j$, thus the above homomorphisms are equal. Thus, we have $c \circ u = \mathsf{id}_C$, hence $c \circ u \sim \mathsf{id}_C$ and also $c \circ u \equiv \mathsf{id}_C$. The latter together with θ implies that for any a, t, we have $u(c(a,t)) \equiv (a,t)$. Thus $u \circ c \sim \mathsf{id}_{\Sigma_{x:A}B(x) \to C}$. $\square$

COROLLARY 3.32. *Over $\mathcal{H}_{\text{ext}}$, given $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$ and $a_1, a_2 : A$, $t_1 : B(a_1) \to \mathsf{W}$, $t_2 : B(a_2) \to \mathsf{W}$ we have*

$$\mathsf{sup}(a_1,t_1) = \mathsf{sup}(a_2,t_2) \quad \simeq \quad (a_1,t_1) = (a_2,t_2)$$

PROOF. By Lem. 3.31 and Cor. 3.29, the structure map sup defines an equivalence between $\Sigma_{x:A}B(x) \to \mathsf{W}$ and W. The rest follows from Lem. **??**[3]. $\square$ EdN:3

[3] EDNOTE: Which says that $a = b$ is equivalent to $f(a) = f(b)$ if f is an equivalence

4. INDUCTIVE TYPES IN HOMOTOPY TYPE THEORY

We now investigate inductive types with propositional computation rules in the setting of homotopy type theory. We start by revisiting the type 2 of Boolean truth values.

4.1. The type 2

Although the rules for the type 2 as presented in section 3.1 are the natural ones to consider in the setting of homotopy type theory, they do not imply a strict universal property. For example, given a type $C : \mathcal{U}_i$ and elements $c_0, c_1 : C$, the function $\lambda_{x:2}\mathsf{2rec}_{\mathcal{U}_i}(C, c_0, c_1, x) : 2 \to C$ cannot, in general, be shown to be definitionally unique
EdN:4 among the functions $f : 2 \to C$ with the property that $f(0) \equiv c_0 : C$ and $f(1) \equiv c_1 : C$.[4] The best that one can do is to show that it is unique among all such maps up to a path, which itself is unique up to a higher path, which in turn is unique up to a yet higher path, etc. This sort of "homotopy" ω-universality, which apparently involves infinitely much data, can nonetheless be captured directly within the system of type theory (without resorting to coinduction) using ideas from higher category theory. To obtain a full characterization of the type 2 using this alternate universal condition, we relax the computation rules from section 3.1 to involve propositional, rather than definitional, equality:

— Propositional 2-computation rules.

$$\frac{x : 2 \vdash E(x) : \mathcal{U}_i \qquad e_0 : E(0) \qquad e_1 : E(1)}{\begin{cases} \mathsf{2ind\text{-}comp0}_{\mathcal{U}_i}(x.E, e_0, e_1) : \mathsf{2ind}_{\mathcal{U}_i}(0) = e_0 \,, \\ \mathsf{2ind\text{-}comp1}_{\mathcal{U}_i}(x.E, e_0, e_1) : \mathsf{2ind}_{\mathcal{U}_i}(1) = e_1 \,. \end{cases}}$$

— Simple propositional 2-computation rules.

$$\frac{C : \mathcal{U}_i \qquad c_0 : C \qquad c_1 : C \qquad b : 2}{\begin{cases} \mathsf{2rec\text{-}comp0}_{\mathcal{U}_i}(C, c_0, c_1) : \mathsf{2rec}_{\mathcal{U}_i}(0) = c_0 \,, \\ \mathsf{2rec\text{-}comp1}_{\mathcal{U}_i}(C, c_0, c_1) : \mathsf{2rec}_{\mathcal{U}_i}(1) = c_1 \,. \end{cases}}$$

As discussed above, the uniqueness principles can still be shown to hold, albeit in a propositional form:

— Propositional 2-uniqueness principle.

$$\frac{\begin{array}{ccc} b : 2 \quad x : 2 \vdash E(x) : \mathcal{U}_i & e_0 : E(0) & e_1 : E(1) \\ x : 2 \vdash f(x) : E(x) & \gamma_0 : f(0) = e_0 & \gamma_1 : f(1) = e_1 \\ x : 2 \vdash g(x) : E(x) & \delta_0 : g(0) = e_0 & \delta_1 : g(1) = e_1 \end{array}}{\mathsf{2ind\text{-}uniq}_{\mathcal{U}_i}(x.E, e_0, e_1, x.f, \gamma_0, \gamma_1, x.g, \delta_0, \delta_1, b) : f(b) = g(b)}$$

— Simple propositional 2-uniqueness principle.

$$\frac{\begin{array}{ccc} b : 2 \quad C : \mathcal{U}_i & c_0 : C & c_1 : C \\ x : 2 \vdash f(x) : C & \gamma_0 : f(0) = c_0 & \gamma_1 : f(1) \equiv c_1 \\ x : 2 \vdash g(x) : C & \delta_0 : g(0) = c_0 & \delta_1 : g(1) = c_1 \end{array}}{\mathsf{2rec\text{-}uniq}_{\mathcal{U}_i}(C, c_0, c_1, x.f, \gamma_0, \gamma_1, x.g, \delta_0, \delta_1, b) : f(b) = g(b)}$$

The witness terms will be shortened to $\mathsf{2ind\text{-}uniq}(b)$, $\mathsf{2rec\text{-}uniq}(b)$ where appropriate. In showing the above laws, we proceed analogously to the extensional case: we use the

[4] EDNOTE: Do we need a reference for this?

induction principle with the type family $x \mapsto f(x) =_{E(x)} g(x)$ and for the cases $x :\equiv 0$, $x :\equiv 1$ we supply the paths $\gamma_0 \cdot \delta_0^{-1}$ and $\gamma_1 \cdot \delta_1^{-1}$ respectively. This gives us a pointwise propositional equality between f and g as desired. The simple uniqueness principle follows at once from the dependent one.

Unlike in the extensional setting, however, $\mathsf{2ind\text{-}uniq}(b)$ will generally not be an identity path. The natural question to ask is how $\mathsf{2ind\text{-}uniq}(-)$ behaves on the constructors 0, 1. Since $\mathsf{2ind\text{-}uniq}(b)$ was constructed using the 2-induction principle, we invoke the corresponding (propositional) computation rules to get the following coherence rules:

— Propositional 2-coherence principles.

$$\frac{\begin{array}{ccc} x : \mathsf{2} \vdash E(x) : \mathcal{U}_i & e_0 : E(0) & e_1 : E(1) \\ x : \mathsf{2} \vdash f(x) : E(x) & \gamma_0 : f(0) = e_0 & \gamma_1 : f(1) = e_1 \\ x : \mathsf{2} \vdash g(x) : E(x) & \delta_0 : g(0) = e_0 & \delta_1 : g(1) = e_1 \end{array}}{\begin{cases} \mathsf{2ind\text{-}coh0}_{\mathcal{U}_i}(x.E, e_0, e_1, x.f, \gamma_0, \gamma_1, x.g, \delta_0, \delta_1) : \mathsf{2ind\text{-}uniq}(0) = \gamma_0 \cdot \delta_0^{-1}, \\ \mathsf{2ind\text{-}coh1}_{\mathcal{U}_i}(x.E, e_0, e_1, x.f, \gamma_0, \gamma_1, x.g, \delta_0, \delta_1) : \mathsf{2ind\text{-}uniq}(1) = \gamma_1 \cdot \delta_1^{-1}. \end{cases}}$$

— Simple propositional 2-coherence principles.

$$\frac{\begin{array}{ccc} C : \mathcal{U}_i & c_0 : C & c_1 : C \\ x : \mathsf{2} \vdash f(x) : C & \gamma_0 : f(0) = c_0 & \gamma_1 : f(1) = c_1 \\ x : \mathsf{2} \vdash g(x) : C & \delta_0 : g(0) = c_0 & \delta_1 : g(1) = c_1 \end{array}}{\begin{cases} \mathsf{2rec\text{-}coh0}_{\mathcal{U}_i}(C, c_0, c_1, x.f, \gamma_0, \gamma_1, x.g, \delta_0, \delta_1) : \mathsf{2rec\text{-}uniq}(0) = \gamma_0 \cdot \delta_0^{-1}, \\ \mathsf{2rec\text{-}coh1}_{\mathcal{U}_i}(C, c_0, c_1, x.f, \gamma_0, \gamma_1, x.g, \delta_0, \delta_1) : \mathsf{2rec\text{-}uniq}(1) = \gamma_1 \cdot \delta_1^{-1}. \end{cases}}$$

This motivates the following definitions.

Definition 4.1. For $\mathcal{X} : \mathsf{2\text{-}Alg}_{\mathcal{U}_i}$, $\mathcal{Y} : \mathsf{2\text{-}Alg}_{\mathcal{U}_j}$ and homomorphisms $\mu, \nu : \mathsf{2\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}$, define the type of *2-cells* between μ and ν by

$$\begin{aligned} &\mathsf{2\text{-}Cell}\ (C, c_0, c_1)\ (D, d_0\ d_1)\ (f, \gamma_0, \gamma_1)\ (g, \delta_0, \delta_1) :\equiv \\ &\quad \mathsf{2\text{-}Fib\text{-}Hom}\ (C, c_0, c_1)\ (f \sim g, \gamma_0 \cdot \delta_0^{-1}, \gamma_1 \cdot \delta_1^{-1}) \end{aligned}$$

Definition 4.2. For $\mathcal{X} : \mathsf{2\text{-}Alg}_{\mathcal{U}_i}$, $\mathcal{Y} : \mathsf{2\text{-}Fib\text{-}Alg}_{\mathcal{U}_j}\ \mathcal{X}$ and fibered homomorphisms $\mu, \nu : \mathsf{2\text{-}Fib\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}$, define the type of *fibered 2-cells* between μ and ν by

$$\begin{aligned} &\mathsf{2\text{-}Fib\text{-}Cell}\ (C, c_0, c_1)\ (E, e_0\ e_1)\ (f, \gamma_0, \gamma_1)\ (g, \delta_0, \delta_1) :\equiv \\ &\quad \mathsf{2\text{-}Fib\text{-}Hom}\ (C, c_0, c_1)\ (f \sim g, \gamma_0 \cdot \delta_0^{-1}, \gamma_1 \cdot \delta_1^{-1}) \end{aligned}$$

For brevity, we will often leave out the first two arguments.

We also need to update some of our previous terminology: it is a misnomer to call an algebra $\mathcal{X}$ *initial* if the type of homomorphisms from $\mathcal{X}$ to any other algebra $\mathcal{Y}$ is contractible, since a contractible type may contain more than one element. Instead, an algebra with this property will be called *homotopy-initial*; by Lem. **??** the contractibility requirement exactly captures the notion that there exists a homomorphism which is unique up to a path, which itself is unique up to a higher path, and so on.

Definition 4.3. An algebra $\mathcal{X} : \mathsf{2\text{-}Alg}_{\mathcal{U}_i}$ is *homotopy-initial* on a universe $\mathcal{U}_j$ if for any algebra $\mathcal{Y} : \mathsf{2\text{-}Alg}_{\mathcal{U}_j}$, the type of 2-homomorphisms between $\mathcal{X}$ and $\mathcal{Y}$ is contractible:

$$\mathsf{is\text{-}2\text{-}hinit}_{\mathcal{U}_j}(\mathcal{X}) :\equiv \Pi_{(\mathcal{Y} : \mathsf{2\text{-}Alg}_{\mathcal{U}_j})} \mathsf{is\text{-}contr}(\mathsf{2\text{-}Hom}\ \mathcal{X}\ \mathcal{Y})$$

In certain respects, algebras and algebra homomorphisms behave much like objects and morphisms in a category:

Definition 4.4. For any $\mathcal{X} : \mathsf{2\text{-}Alg}_{\mathcal{U}_i}$ there is a designated *identity* homomorphism from $\mathcal{X}$ to itself, defined by

$$\mathsf{2\text{-}Id\text{-}Hom}\ (C, c_0, c_1) :\equiv (\mathsf{id}_C, \mathsf{refl}(c_0), \mathsf{refl}(c_1))$$

We often denote this homomorphism by $1_{\mathcal{X}}$.

Definition 4.5. For algebras $\mathcal{X} : \mathsf{2\text{-}Alg}_{\mathcal{U}_i}, \mathcal{Y} : \mathsf{2\text{-}Alg}_{\mathcal{U}_j}, \mathcal{Z} : \mathsf{2\text{-}Alg}_{\mathcal{U}_k}$ and homomorphisms $\mu : \mathsf{2\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}$, $\nu : \mathsf{2\text{-}Hom}\ \mathcal{Y}\ \mathcal{Z}$, the *composition* of μ and ν is a homomorphism from $\mathcal{X}$ to $\mathcal{Z}$ defined by

$$\begin{aligned}&\mathsf{2\text{-}Comp\text{-}Hom}\ (C, c_0, c_1)\ (D, d_0, d_1)\ (E, e_0, e_1)\ (f, \gamma_0, \gamma_1)\ (g, \delta_0, \delta_1) :\equiv \\ &\quad \big(g \circ f, g(\gamma_0) \cdot \delta_0, g(\gamma_1) \cdot \delta_1\big)\end{aligned}$$

We often leave out the first three arguments and denote the composition by $\nu \circ \mu$.

Definition 4.6. For $\mathcal{X} : \mathsf{2\text{-}Alg}_{\mathcal{U}_i}$, $\mathcal{Y} : \mathsf{2\text{-}Alg}_{\mathcal{U}_j}$ we define the type of *isomorphisms* between $\mathcal{X}$ and $\mathcal{Y}$ as

$$\mathsf{2\text{-}Alg\text{-}Iso}\ \mathcal{X}\ \mathcal{Y} :\equiv \Sigma_{(\rho:\mathsf{2\text{-}Hom}\ \mathcal{X}\ \mathcal{Y})}\Big(\Sigma_{(\mu:\mathsf{2\text{-}Hom}\ \mathcal{Y}\ \mathcal{X})}\mu \circ \rho = 1_{\mathcal{X}}\Big) \times \Big(\Sigma_{(\nu:\mathsf{2\text{-}Hom}\ \mathcal{Y}\ \mathcal{X})}\rho \circ \nu = 1_{\mathcal{Y}}\Big)$$

LEMMA 4.7. *For $\mathcal{X} : \mathsf{2\text{-}Alg}_{\mathcal{U}_i}$, $\mathcal{Y} : \mathsf{2\text{-}Fib\text{-}Alg}_{\mathcal{U}_j}\ \mathcal{X}$ and $\mu, \nu : \mathsf{2\text{-}Fib\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}$, the path space $\mu = \nu$ is equivalent to the space of fibered 2-cells between μ and ν:*

$$\mu = \nu \ \ \simeq \ \ \mathsf{2\text{-}Fib\text{-}Cell}\ \mu\ \nu$$

PROOF. Let algebras $(C, c_0, c_1) : \mathsf{2\text{-}Alg}_{\mathcal{U}_i}$, $(E, e_0, e_1) : \mathsf{2\text{-}Fib\text{-}Alg}_{\mathcal{U}_j}\ (C, c_0, c_1)$ and homomorphisms $(f, \gamma_0, \gamma_1), (g, \delta_0, \delta_1) : \mathsf{2\text{-}Fib\text{-}Hom}\ (C, c_0, c_1)\ (E, e_0, e_1)$ be given. We have

$$\begin{aligned}
&(f, \gamma_0, \gamma_1) = (g, \delta_0, \delta_1) && \simeq \\
&\Sigma_{\alpha:f=g}\Big((\gamma_0, \gamma_1) = \mathsf{tr}^{-1}_{h \mapsto h(c_0)=d_0 \times h(c_1)=d_1}(\alpha)\ (\delta_0, \delta_1)\Big) && \simeq \\
&\Sigma_{\alpha:f=g}\Big((\gamma_0, \gamma_1) = \big({}^{=}\mathsf{Eq}^{\Pi}(\alpha, c_0) \cdot \delta_0, {}^{=}\mathsf{Eq}^{\Pi}(\alpha, c_1) \cdot \delta_1\big)\Big) && \simeq \\
&\Sigma_{\alpha:f=g}\Big(\gamma_0 = {}^{=}\mathsf{Eq}^{\Pi}(\alpha, c_0) \cdot \delta_0\Big) \times \Big(\gamma_1 = {}^{=}\mathsf{Eq}^{\Pi}(\alpha, c_1) \cdot \delta_1\Big) && \simeq \\
&\Sigma_{\alpha:f\sim g}\Big(\gamma_0 = \alpha(c_0) \cdot \delta_0\Big) \times \Big(\gamma_1 = \alpha(c_1) \cdot \delta_1\Big) && \simeq \\
&\Sigma_{\alpha:f\sim g}\Big(\alpha(c_0) = \gamma_0 \cdot \delta_0^{-1}\Big) \times \Big(\alpha(c_1) = \gamma_1 \cdot \delta_1^{-1}\Big) && \equiv \\
&\mathsf{2\text{-}Fib\text{-}Cell}\ (f, \gamma_0, \gamma_1)\ (g, \delta_0, \delta_1)
\end{aligned}$$

□

COROLLARY 4.8. *For $\mathcal{X} : \mathsf{2\text{-}Alg}_{\mathcal{U}_i}$, $\mathcal{Y} : \mathsf{2\text{-}Alg}_{\mathcal{U}_j}$ and $\mu, \nu : \mathsf{2\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}$, the path space $\mu = \nu$ is equivalent to the space of 2-cells between μ and ν:*

$$\mu = \nu \ \ \simeq \ \ \mathsf{2\text{-}Cell}\ \mu\ \nu$$

LEMMA 4.9. *For $\mathcal{X}, \mathcal{Y} : \mathsf{2\text{-}Alg}_{\mathcal{U}_i}$, the path space $\mathcal{X} = \mathcal{Y}$ is equivalent to the space of isomorphisms between $\mathcal{X}$ and $\mathcal{Y}$:*

$$\mathcal{X} = \mathcal{Y} \ \ \simeq \ \ \mathsf{2\text{-}Alg\text{-}Iso}\ \mathcal{X}\ \mathcal{Y}$$

PROOF. Let algebras $(C, c_0, c_1), (D, d_0, d_1) : \text{2-Alg}_{\mathcal{U}_i}$ be given. We have

$$\begin{aligned}
&\text{2-Alg-Iso } (C, c_0, c_1)\ (D, d_0, d_1) && \equiv \\
&\Sigma_{\rho:\left(\Sigma_{f:C\to D} f(c_0)=d_0 \times f(c_1)=d_1\right)} \left(\Sigma_{\mu:\left(\Sigma_{g:D\to C} g(d_0)=c_0\times g(d_1)=c_1\right)} \mu\circ\rho = 1_{(C,c_0,c_1)}\right) \times \\
&\qquad \left(\Sigma_{\nu:\left(\Sigma_{h:D\to C} h(d_0)=c_0\times g(d_1)=c_1\right)} \rho\circ\nu = 1_{(D,d_0,d_1)}\right) && \simeq \\
&\Sigma_{f:C\to D}\Sigma_{\epsilon_0:f(c_0)=d_0}\Sigma_{\epsilon_1:f(c_1)=d_1} \left(\Sigma_{g:D\to C}\Sigma_{\gamma_0:g(d_0)=c_0}\Sigma_{\gamma_1:g(d_1)=c_1} P_{f,\epsilon_0,\epsilon_1,g,\gamma_0,\gamma_1}\right) \times \\
&\qquad \left(\Sigma_{h:D\to C}\Sigma_{\delta_0:h(d_0)=c_0}\Sigma_{\delta_1:h(d_1)=c_1} Q_{f,\epsilon_0,\epsilon_1,h,\delta_0,\delta_1}\right)
\end{aligned}$$

where

$$\begin{aligned}
P_{f,\epsilon_0,\epsilon_1,g,\gamma_0,\gamma_1} &:\equiv \left(g\circ f, \mathsf{ap}_g(\epsilon_0)\cdot\gamma_0, \mathsf{ap}_g(\epsilon_1)\cdot\gamma_1\right) = \left(\mathsf{id}_C, \mathsf{refl}_{c_0}, \mathsf{refl}_{c_1}\right) \\
Q_{f,\epsilon_0,\epsilon_1,h,\delta_0,\delta_1} &:\equiv \left(f\circ h, \mathsf{ap}_f(\delta_0)\cdot\epsilon_0, \mathsf{ap}_f(\delta_1)\cdot\epsilon_1\right) = \left(\mathsf{id}_D, \mathsf{refl}_{d_0}, \mathsf{refl}_{d_1}\right)
\end{aligned}$$

Now we have

$$\begin{aligned}
&P_{f,\epsilon_0,\epsilon_1,g,\gamma_0,\gamma_1} && \equiv \\
&\left(g\circ f, \mathsf{ap}_g(\epsilon_0)\cdot\gamma_0, \mathsf{ap}_g(\epsilon_1)\cdot\gamma_1\right) = \left(\mathsf{id}_C, \mathsf{refl}_{c_0}, \mathsf{refl}_{c_1}\right) && \simeq \\
&\Sigma_{\alpha:g\circ f=\mathsf{id}_C} \left(\mathsf{ap}_g(\epsilon_0)\cdot\gamma_0, \mathsf{ap}_g(\epsilon_1)\cdot\gamma_1\right) = \mathsf{tr}^{-1}_{i\mapsto i(c_0)=c_0\times i(c_1)=c_1}(\alpha)\ \left(\mathsf{refl}_{c_0}, \mathsf{refl}_{c_1}\right) && \simeq \\
&\Sigma_{\alpha:g\circ f=\mathsf{id}_C} \left(\mathsf{ap}_g(\epsilon_0)\cdot\gamma_0, \mathsf{ap}_g(\epsilon_1)\cdot\gamma_1\right) = \left({}^{=}\mathsf{Eq}^{\Pi}(\alpha, c_0)\cdot\mathsf{refl}_{c_0}, {}^{=}\mathsf{Eq}^{\Pi}(\alpha, c_1)\cdot\mathsf{refl}_{c_1}\right) && \simeq \\
&\Sigma_{\alpha:g\circ f=\mathsf{id}_C} \left(\mathsf{ap}_g(\epsilon_0)\cdot\gamma_0 = {}^{=}\mathsf{Eq}^{\Pi}(\alpha, c_0)\cdot\mathsf{refl}_{c_0}\right) \times \left(\mathsf{ap}_g(\epsilon_1)\cdot\gamma_1 = {}^{=}\mathsf{Eq}^{\Pi}(\alpha, c_1)\cdot\mathsf{refl}_{c_1}\right) && \simeq \\
&\Sigma_{\alpha:g\circ f\sim\mathsf{id}_C} \left(\mathsf{ap}_g(\epsilon_0)\cdot\gamma_0 = \alpha(c_0)\cdot\mathsf{refl}_{c_0}\right) \times \left(\mathsf{ap}_g(\epsilon_1)\cdot\gamma_1 = \alpha(c_1)\cdot\mathsf{refl}_{c_1}\right) && \simeq \\
&\Sigma_{\alpha:g\circ f\sim\mathsf{id}_C} \left(\alpha(c_0) = \mathsf{ap}_g(\epsilon_0)\cdot\gamma_0\right) \times \left(\alpha(c_1) = \mathsf{ap}_g(\epsilon_1)\cdot\gamma_1\right)
\end{aligned}$$

and analogously

$$Q_{f,\epsilon_0,\epsilon_1,h,\delta_0,\delta_1} \simeq \Sigma_{\beta:f\circ h\sim\mathsf{id}_D} \left(\beta(d_0) = \mathsf{ap}_f(\delta_0)\cdot\epsilon_0\right) \times \left(\beta(d_1) = \mathsf{ap}_f(\delta_1)\cdot\epsilon_1\right)$$

Thus we can express 2-Alg-Iso $(C, c_0, c_1)\ (D, d_0, d_1)$ equivalently as the type

$$\begin{aligned}
&\Sigma_{f:C\to D}\Sigma_{\epsilon_0:f(c_0)=d_0}\Sigma_{\epsilon_1:f(c_1)=d_1}\Sigma_{g:D\to C}\Sigma_{\alpha:g\circ f\sim\mathsf{id}_C}\Sigma_{h:D\to C}\Sigma_{\beta:f\circ h\sim\mathsf{id}_D} \\
&\left(\Sigma_{\gamma_0:g(d_0)=c_0}\alpha(c_0) = \mathsf{ap}_g(\epsilon_0)\cdot\gamma_0\right) \times \left(\Sigma_{\gamma_1:g(d_1)=c_1}\alpha(c_1) = \mathsf{ap}_g(\epsilon_1)\cdot\gamma_1\right) \times \\
&\left(\Sigma_{\delta_0:h(d_0)=c_0}\beta(d_0) = \mathsf{ap}_f(\delta_0)\cdot\epsilon_0\right) \times \left(\Sigma_{\delta_1:h(d_1)=c_1}\beta(d_1) = \mathsf{ap}_f(\delta_1)\cdot\epsilon_1\right)
\end{aligned}$$

Now we have

$$\Sigma_{\gamma_0:g(d_0)=c_0}\left(\alpha(c_0) = \mathsf{ap}_g(\epsilon_0)\cdot\gamma_0\right) \quad\simeq\quad \Sigma_{\gamma_0:g(d_0)=c_0}\left(\gamma_0 = \mathsf{ap}_g(\epsilon_0)^{-1}\cdot\alpha(c_0)\right) \quad\simeq\quad 1$$

by Lem. ??[5]. Similarly, $\Sigma_{\gamma_1:g(d_1)=c_1}\left(\alpha(c_1) = \mathsf{ap}_g(\epsilon_1)\cdot\gamma_1\right)$ is equivalent to 1. We also have EdN:5

$$\Sigma_{\delta_0:h(d_0)=c_0}\left(\beta(d_0) = \mathsf{ap}_f(\delta_0)\cdot\epsilon_0\right) \quad\simeq\quad \Sigma_{\delta_0:h(d_0)=c_0}\left(\mathsf{ap}_f(\delta_0) = \beta(d_0)\cdot\epsilon_0^{-1}\right) \quad\simeq\quad 1$$

[5] EDNOTE: Which says is-contr$(\Sigma_{x:A} x = y)$

EdN:6 by Lem. **??**[6], **??**[7], and the fact that g, α, h, β make f into an equivalence. Similarly,
EdN:7 $\Sigma_{\delta_1 : h(d_1) = c_1}\Big(\beta(d_1) = \mathsf{ap}_f(\delta_1) \cdot \epsilon_1\Big)$ is equivalent to 1. Thus, we have

$$
\begin{aligned}
&\text{2-Alg-Iso } (C, c_0, c_1)\ (D, d_0, d_1) && \simeq \\
&\Sigma_{f:C\to D}\Sigma_{g:D\to C}\Sigma_{\alpha:g\circ f\sim \mathsf{id}_C}\Sigma_{h:D\to C}\Sigma_{\beta:f\circ h\sim \mathsf{id}_D}\Big(f(c_0) = d_0\Big) \times \Big(f(c_1) = d_1\Big) && \simeq \\
&\Sigma_{f:C\to D}\mathsf{is\text{-}equiv}(f) \times \Big(f(c_0) = d_0\Big) \times \Big(f(c_1) = d_1\Big) && \simeq \\
&\Sigma_{p:C\simeq D}\Big(\mathsf{fst}(p)\ c_0 = d_0\Big) \times \Big(\mathsf{fst}(p)\ c_1 = d_1\Big) && \simeq \\
&\Sigma_{p:C\simeq D}\Big(\mathsf{fst}(p)\ c_0, \mathsf{fst}(p)\ c_1\Big) = (d_0, d_1) && \simeq \\
&\Sigma_{p:C=D}\Big(\mathsf{fst}({}^{=}\mathsf{Eq}^{\simeq}(p))\ c_0, \mathsf{fst}({}^{=}\mathsf{Eq}^{\simeq}(p))\ c_1\Big) = (d_0, d_1) && \simeq \\
&\Sigma_{p:C=D}\Big(\mathsf{tr}_{X\mapsto X\times X}(p)\ (c_0, c_1) = (d_0, d_1)\Big) && \simeq \\
&(C, c_0, c_1) = (D, d_0, d_1)
\end{aligned}
$$

□

As desired, the analogues of the lemmas from section 3.1 still hold in the setting of homotopy type theory:

LEMMA 4.10. *In $\mathcal{H}$, if an algebra $\mathcal{X}$: 2-Alg$_{\mathcal{U}_i}$ satisfies the induction principle on the universe $\mathcal{U}_j$, it also satisfies the induction uniqueness principle on $\mathcal{U}_j$. In other words, we have*

$$\mathsf{has\text{-}2\text{-}ind}_{\mathcal{U}_j}(\mathcal{X}) \quad\to\quad \mathsf{has\text{-}2\text{-}ind\text{-}uniq}_{\mathcal{U}_j}(\mathcal{X})$$

PROOF. Let an algebra (C, c_0, c_1) : 2-Alg$_{\mathcal{U}_i}$ be given. To prove the induction uniqueness principle, take any algebra (E, e_0, e_1) : 2-Fib-Alg$_{\mathcal{U}_j}$ (C, c_0, c_1) and homomorphisms $(f, \gamma_0, \gamma_1), (g, \delta_0, \delta_1)$: 2-Fib-Hom (C, c_0, c_1) (E, e_0, e_1). By Lem. 4.7, to show $(f, \gamma_0, \gamma_1) = (g, \delta_0, \delta_1)$ it suffices to exhibit a fibered 2-cell between (f, γ_0, γ_1) and (g, δ_0, δ_1). For this we use the induction principle with the fibered algebra $\big(f \sim g, \gamma_0 \cdot \delta_0^{-1}, \gamma_1 \cdot \delta_1^{-1}\big)$ and we are done. □

COROLLARY 4.11. *In $\mathcal{H}$, if an algebra $\mathcal{X}$: 2-Alg$_{\mathcal{U}_i}$ satisfies the induction principle on the universe $\mathcal{U}_j$, it is homotopy-initial on $\mathcal{U}_j$. In other words, we have*

$$\mathsf{has\text{-}2\text{-}ind}_{\mathcal{U}_j}(\mathcal{X}) \quad\to\quad \mathsf{is\text{-}2\text{-}hinit}_{\mathcal{U}_j}(\mathcal{X})$$

LEMMA 4.12. *In $\mathcal{H}$, if an algebra $\mathcal{X}$: 2-Alg$_{\mathcal{U}_i}$ satisfies the recursion and recursion uniqueness principles on the universe $\mathcal{U}_j$ and $j \geq i$, then it satisfies the induction principle on $\mathcal{U}_j$. In other words, we have*

$$\mathsf{has\text{-}2\text{-}rec}_{\mathcal{U}_j}(\mathcal{X}) \times \mathsf{has\text{-}2\text{-}rec\text{-}uniq}_{\mathcal{U}_j}(\mathcal{X}) \quad\to\quad \mathsf{has\text{-}2\text{-}ind}_{\mathcal{U}_j}(\mathcal{X})$$

provided $j \geq i$.

PROOF. Let algebras (C, c_0, c_1) : 2-Alg$_{\mathcal{U}_i}$ and (E, e_0, e_1) : 2-Fib-Alg$_{\mathcal{U}_j}$ (C, c_0, c_1) be given. We use the recursion principle with the algebra $\big(\Sigma_{x:C}E(x), (c_0, e_0), (c_1, e_1)\big)$. We note that the carrier type belongs to $\mathcal{U}_j$ as $i \leq j$. This gives us a homomorphism

[6] EDNOTE: Which says is-contr$(\Sigma_{x:A}f(x) =_B y)$ if $f : A \to B$ is an equivalence
[7] EDNOTE: Which says $\mathsf{ap}_f : x =_A y \to f(x) =_B f(y)$ is equivalence if $f : A \to B$ is

(u, θ_0, θ_1), where $u : C \to \Sigma_{x:C} E(x)$, $\theta_0 : u(c_0) = (c_0, e_0)$, and $\theta_1 : u(c_1) = (c_1, e_1)$. We can now form two homomorphisms

$$\Big(\mathsf{fst} \circ u, \mathsf{fst}({}^{=}\mathsf{Eq}^{\Sigma}(\theta_0)), \mathsf{fst}({}^{=}\mathsf{Eq}^{\Sigma}(\theta_1))\Big), \Big(\mathsf{id}_C, \mathsf{refl}_{c_0}, \mathsf{refl}_{c_1}\Big) : \text{2-Hom } (C, c_0, c_1)\ (C, c_0, c_1)$$

The recursion uniqueness principle tells us that these homomorphisms are equal. By Cor. 4.8 this means there exists a 2-cell (α, η_0, η_1) between them, i.e., we have

$$\alpha : \mathsf{fst} \circ u \sim \mathsf{id}_C$$
$$\eta_0 : \alpha(c_0) = \mathsf{fst}({}^{=}\mathsf{Eq}^{\Sigma}(\theta_0)) \cdot \mathsf{refl}_{c_0}$$
$$\eta_1 : \alpha(c_1) = \mathsf{fst}({}^{=}\mathsf{Eq}^{\Sigma}(\theta_1)) \cdot \mathsf{refl}_{c_1}$$

We can thus define the desired fibered homomorphism as

$$\Big(x \mapsto \mathsf{tr}_E(\alpha(x))\ \mathsf{snd}(u(x)), \phi_0, \phi_1\Big) : \text{2-Fib-Hom } (C, c_0, c_1)\ (E, e_0, e_1)$$

where ϕ_0 is the path

$$\begin{array}{c}
\mathsf{tr}_E\big(\alpha(c_0)\big)\ \mathsf{snd}(u(c_0)) \\
\Big| \ \text{via } \eta_0 \\
\mathsf{tr}_E\big(\mathsf{fst}({}^{=}\mathsf{Eq}^{\Sigma}(\theta_0)) \cdot \mathsf{refl}_{c_0}\big)\ \mathsf{snd}(u(c_0)) \\
\Big| \\
\mathsf{tr}_E\big(\mathsf{fst}({}^{=}\mathsf{Eq}^{\Sigma}(\theta_0))\big)\ \mathsf{snd}(u(c_0)) \\
\Big| \ \mathsf{snd}({}^{=}\mathsf{Eq}^{\Sigma}(\theta_0)) \\
e_0
\end{array}$$

and ϕ_1 is defined analogously. □

COROLLARY 4.13. *In $\mathcal{H}$, the following conditions on an algebra $\mathcal{X}$: 2-Alg$_{\mathcal{U}_i}$ are equivalent:*

(*1*) *$\mathcal{X}$ satisfies the induction principle on the universe $\mathcal{U}_j$*
(*2*) *$\mathcal{X}$ satisfies the recursion and recursion uniqueness principles on the universe $\mathcal{U}_j$*
(*3*) *$\mathcal{X}$ is homotopy-initial on the universe $\mathcal{U}_j$*

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

$$\text{has-2-ind}_{\mathcal{U}_j}(\mathcal{X}) \quad \simeq \quad \text{has-2-rec}_{\mathcal{U}_j}(\mathcal{X}) \times \text{has-2-rec-uniq}_{\mathcal{U}_j}(\mathcal{X}) \quad \simeq \quad \text{is-2-hinit}_{\mathcal{U}_j}(\mathcal{X})$$

provided $j \geq i$. Furthermore, all 3 conditions are mere propositions.

PROOF. Conditions (1) and (3) are mere propositions by Lem. **??** (for the latter), Lem. **??**, 4.10 (for the former), and the fact that a family of mere propositions is itself a mere proposition. □

Furthermore, we have the following corollary which does *not* have an analogue in the extensional case:

COROLLARY 4.14. *In $\mathcal{H}$, any two algebras $\mathcal{X}, \mathcal{Y}$: 2-Alg$_{\mathcal{U}_i}$ which are homotopy-initial on $\mathcal{U}_i$ are equal:*

$$\text{is-2-hinit}_{\mathcal{U}_i}(\mathcal{X}) \to \text{is-2-hinit}_{\mathcal{U}_i}(\mathcal{Y}) \to \mathcal{X} = \mathcal{Y}$$

PROOF. By Lem. 4.9 it suffices to construct an isomorphism between $\mathcal{X}$ and $\mathcal{Y}$. Since $\mathcal{X}$ and $\mathcal{Y}$ are both homotopy-initial on $\mathcal{U}_i$, there exist homomorphisms $\mu : \text{2-Hom } \mathcal{X}\ \mathcal{Y}$ and $\nu : \text{2-Hom } \mathcal{Y}\ \mathcal{X}$. Again by homotopy-initiality, we have $\nu \circ \mu = 1_{\mathcal{X}}$ and $\mu \circ \nu = 1_{\mathcal{Y}}$, which gives us the desired isomorphism. □

We can thus characterize the type 2 using the universal property of homotopy-initiality as follows.

COROLLARY 4.15. *In $\mathcal{H}$ extended with the type 2, the algebra $(2, 0, 1) : \text{2-Alg}_{\mathcal{U}_0}$ is homotopy-initial on any universe $\mathcal{U}_j$.*

COROLLARY 4.16. *In $\mathcal{H}$ extended with an algebra $\mathcal{X} : \text{2-Alg}_{\mathcal{U}_0}$ which is homotopy-initial on any universe $\mathcal{U}_j$, the type 2 with propositional computation rules is definable.*

PROOF. We have an algebra $\cdot \vdash \mathcal{X} : \text{2-Alg}_{\mathcal{U}_0}$ such that for any j, there exists a term $\cdot \vdash h_j : \text{is-2-hinit}_{\mathcal{U}_j}(\mathcal{X})$. Since the requirement $j \geq 0$ always holds, Cor. 4.13 implies that for any j, we have a term $\cdot \vdash r_j : \text{has-2-ind}_{\mathcal{U}_j}(\mathcal{X})$. This implies that the type 2 with propositional computation rules is definable. □

Corollaries 4.16 and 4.15 are the analogue in Homotopy Type Theory of the characterization of 2 as a strict coproduct $1 + 1$ in extensional type theory. It makes precise the rough idea that, in intensional type theory, 2 is a kind of homotopy coproduct or weak ω-coproduct in the weak ω-category $\mathcal{C}(\mathcal{H})$ of types, terms, identity terms, higher identity terms, ... It is worth emphasizing that homotopy-initiality is a purely type-theoretic notion; despite having an obvious semantic interpretation, it is formulated in terms of inhabitation of specific, definable types. Indeed, Corollary 4.13 and its proof have been completely formalized in the Coq proof assistant [Awodey et al. 2011].

4.2. W-types

Although it is more elaborate to state (and difficult to prove) owing to the presence of recursively generated data, our main result on W-types is analogous to the foregoing example in the following respect: rather than being strict initial algebras, as in the extensional case, W-types with propositional computation rules are instead homotopy-initial algebras. This fact can again be stated entirely syntactically, as an equivalence between the definability of W-types with propositional computation rules (which we spell out below) and the existence of a suitable family of homotopy-initial algebras. Moreover, as in the simple case of the type 2 above, the proof is again entirely constructive.

The propositional computation rules for W-types are as follows:

— Propositional W-computation rule.

$$\frac{\begin{array}{c} A : \mathcal{U}_i \qquad x : A \vdash B(x) : \mathcal{U}_i \qquad a : A \qquad t : B(a) \to \mathsf{W} \\ w : \mathsf{W} \vdash E(w) : \mathcal{U}_j \qquad x : A, p : B(x) \to \mathsf{W}, r : \Pi_{b:B(x)} E(p\ b) \vdash e(x, p, r) : E(\mathsf{sup}(x, p)) \end{array}}{\mathsf{wind\text{-}comp}_{\mathcal{U}_i}(A, x.B, w.E, x.p.r.e, a, t) : \mathsf{wind}(\mathsf{sup}(a, t)) = e(a, t, \lambda_{b:B(a)} \mathsf{wind}(t\ b))}$$

— Simple propositional W-computation rule.

$$\frac{\begin{array}{c} A : \mathcal{U}_i \qquad x : A \vdash B(x) : \mathcal{U}_i \\ a : A \qquad t : B(a) \to \mathsf{W} \qquad C : \mathcal{U}_j \qquad x : A, r : B(x) \to C \vdash c(x, r) : C \end{array}}{\mathsf{wrec\text{-}comp}_{\mathcal{U}_i}(A, x.B, C, x.r.c, a, t) : \mathsf{wrec}(\mathsf{sup}(a, t)) = c(a, \lambda_{b:B(a)} \mathsf{wrec}(t\ b))}$$

Remark 4.17. One interesting aspect of this group of rules is that, unlike the standard rules for W-types, they are invariant under equivalence, and propositional equality in particular. If $A : \mathcal{U}_i$ and $x : A \vdash B(x) : \mathcal{U}_i$ and we have a type $W : \mathcal{U}_i$ such that

$W \simeq \mathsf{W}^{\mathcal{U}_i}_{x:A} B(x)$, then W can be shown to satisfy the same rules as $\mathsf{W}^{\mathcal{U}_i}_{x:A} B(x)$, in the sense that there are definable terms playing the role of the primitive constants that appear in the rules for $\mathsf{W}^{\mathcal{U}_i}_{x:A} B(x)$.

We again have uniqueness principles, in the propositional form:

— Propositional W-uniqueness principle.

$$\frac{\begin{array}{c} A : \mathcal{U}_i \qquad x : A \vdash B(x) : \mathcal{U}_i \qquad m : \mathsf{W} \\ w : \mathsf{W} \vdash E(w) : \mathcal{U}_j \qquad x : A, p : B(x) \to \mathsf{W}, r : \Pi_{b:B(x)} E(p\ b) \vdash e(x,p,r) : E(\mathsf{sup}(x,p)) \\ w : \mathsf{W} \vdash f(w) : E(w) \qquad w : \mathsf{W} \vdash g(w) : E(w) \\ x : A, p : B(x) \to \mathsf{W} \vdash \gamma(x,p) : f(\mathsf{sup}(x,p)) = e(x,p,\lambda_{b:B(x)} f(p\ b)) \\ x : A, p : B(x) \to \mathsf{W} \vdash \delta(x,p) : g(\mathsf{sup}(x,p)) = e(x,p,\lambda_{b:B(x)} g(p\ b)) \end{array}}{\mathsf{wind\text{-}uniq}_{\mathcal{U}_i}(A, x.B, w.E, x.p.r.e, w.f, x.p.\gamma, w.g, x.p.\delta, m) : f(m) = g(m)}$$

— Simple propositional W-uniqueness principle.

$$\frac{\begin{array}{c} A : \mathcal{U}_i \qquad x : A \vdash B(x) : \mathcal{U}_i \qquad m : \mathsf{W} \qquad C : \mathcal{U}_j \qquad x : A, r : B(x) \to C \vdash c(x,r) : C \\ w : \mathsf{W} \vdash f(w) : C \qquad x : A, p : B(x) \to \mathsf{W} \vdash \gamma(x,p) : f(\mathsf{sup}(x,p)) = c(x, \lambda_{b:B(x)} f(p\ b)) \\ w : \mathsf{W} \vdash g(x) : C \qquad x : A, p : B(x) \to \mathsf{W} \vdash \delta(x,p) : g(\mathsf{sup}(x,p)) = c(x, \lambda_{b:B(x)} g(p\ b)) \end{array}}{\mathsf{wrec\text{-}uniq}_{\mathcal{U}_i}(A, x.B, C, x.r.c, w.f, x.p.\gamma, w.g, x.p.\delta, m) : f(m) = g(m)}$$

The witness terms will be shortened to $\mathsf{wind\text{-}uniq}(m), \mathsf{wrec\text{-}uniq}(m)$ where appropriate. In showing the above laws, we proceed analogously to the extensional case: we use the induction principle with the type family $x \mapsto f(w) =_{E(w)} g(w)$. For this we need to show that for any x, p we have $f(\mathsf{sup}(x,p)) = g(\mathsf{sup}(x,p))$ under the induction hypothesis $h : \Pi_{b:B(x)} f(p\ b) = g(p\ b)$. We can construct this path explicitly as

$$\gamma(x,p) \cdot \mathsf{ap}_{e(x,p,-)}({}^{\Pi}\mathsf{Eq}^{=}(h)) \cdot \delta(x,p)^{-1}$$

The simple uniqueness rule again follows from the dependent one. Invoking the corresponding (propositional) computation rule, we get the following coherence principles for W-types:

— Propositional W-coherence principle.

$$\frac{\begin{array}{c} A : \mathcal{U}_i \qquad x : A \vdash B(x) : \mathcal{U}_i \qquad a : A \qquad t : B(a) \to \mathsf{W} \\ w : \mathsf{W} \vdash E(w) : \mathcal{U}_j \qquad x : A, p : B(x) \to \mathsf{W}, r : \Pi_{b:B(x)} E(p\ b) \vdash e(x,p,r) : E(\mathsf{sup}(x,p)) \\ w : \mathsf{W} \vdash f(w) : E(w) \qquad w : \mathsf{W} \vdash g(w) : E(w) \\ x : A, p : B(x) \to \mathsf{W} \vdash \gamma(x,p) : f(\mathsf{sup}(x,p)) = e(x,p,\lambda_{b:B(x)} f(p\ b)) \\ x : A, p : B(x) \to \mathsf{W} \vdash \delta(x,p) : g(\mathsf{sup}(x,p)) = e(x,p,\lambda_{b:B(x)} g(p\ b)) \end{array}}{\begin{array}{c} \mathsf{wind\text{-}coh}_{\mathcal{U}_i}(A, x.B, w.E, x.p.r.e, w.f, x.p.\gamma, w.g, x.p.\delta, a, t) : \\ \mathsf{wind\text{-}uniq}(\mathsf{sup}(a,t)) = \gamma(a,t) \cdot \mathsf{ap}_{e(a,t,-)}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_{b:B(a)} \mathsf{wind\text{-}uniq}(t\ b))) \cdot \delta(a,t)^{-1} \end{array}}$$

— Simple propositional W-coherence principle.

$$\frac{\begin{array}{c} A : \mathcal{U}_i \qquad x : A \vdash B(x) : \mathcal{U}_i \\ a : A \qquad t : B(a) \to \mathsf{W} \qquad C : \mathcal{U}_j \qquad x : A, r : B(x) \to C \vdash c(x,r) : C \\ w : \mathsf{W} \vdash f(w) : C \qquad x : A, p : B(x) \to \mathsf{W} \vdash \gamma(x,p) : f(\mathsf{sup}(x,p)) = c(x, \lambda_{b:B(x)} f(p\ b)) \\ w : \mathsf{W} \vdash g(x) : C \qquad x : A, p : B(x) \to \mathsf{W} \vdash \delta(x,p) : g(\mathsf{sup}(x,p)) = c(x, \lambda_{b:B(x)} g(p\ b)) \end{array}}{\begin{array}{c} \mathsf{wrec\text{-}coh}_{\mathcal{U}_i}(A, x.B, C, x.r.c, w.f, x.p.\gamma, w.g, x.p.\delta, a, t) : \\ \mathsf{wrec\text{-}uniq}(\mathsf{sup}(a,t)) = \gamma(a,t) \cdot \mathsf{ap}_{c(a,-)}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_{b:B(a)} \mathsf{wrec\text{-}uniq}(t\ b))) \cdot \delta(a,t)^{-1} \end{array}}$$

This motivates the following definitions:

Definition 4.18. For $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, $\mathcal{X}$: $\mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A, B)$, $\mathcal{Y}$: $\mathsf{W\text{-}Alg}_{\mathcal{U}_k}(A, B)$ and homomorphisms μ, ν : $\mathsf{W\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}$, define the type of *2-cells* between μ and ν by

$$\begin{aligned}&\mathsf{W\text{-}Cell}\ (C, c)\ (D, d)\ (f, \gamma)\ (g, \delta) :\equiv \\ &\qquad \mathsf{W\text{-}Fib\text{-}Hom}\ (C, c)\ \Big(f \sim g; x, p, r \mapsto \gamma(x, p) \cdot \mathsf{ap}_{d(x)}({}^{\Pi}\mathsf{Eq}^{=}(r)) \cdot \delta(x, p)^{-1}\Big)\end{aligned}$$

Definition 4.19. For $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, $\mathcal{X}$: $\mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A, B)$, $\mathcal{Y}$: $\mathsf{W\text{-}Fib\text{-}Alg}_{\mathcal{U}_k}(A, B)\ \mathcal{X}$ and fibered homomorphisms μ, ν : $\mathsf{W\text{-}Fib\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}$, define the type of *fibered 2-cells* between μ and ν by

$$\begin{aligned}&\mathsf{W\text{-}Fib\text{-}Cell}\ (C, c)\ (E, e)\ (f, \gamma)\ (g, \delta) :\equiv \\ &\qquad \mathsf{W\text{-}Fib\text{-}Hom}\ (C, c)\ \Big(f \sim g; x, p, r \mapsto \gamma(x, p) \cdot \mathsf{ap}_{e(x,p)}({}^{\Pi}\mathsf{Eq}^{=}(r)) \cdot \delta(x, p)^{-1}\Big)\end{aligned}$$

For brevity, we will often leave out the first two arguments. As expected, we have:

Definition 4.20. Given $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, an algebra $\mathcal{X}$: $\mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A, B)$ is called *homotopy-initial* on a universe $\mathcal{U}_k$ if for any algebra $\mathcal{Y}$: $\mathsf{W\text{-}Alg}_{\mathcal{U}_k}(A, B)$, the type of W-homomorphisms between $\mathcal{X}$ and $\mathcal{Y}$ is contractible:

$$\mathsf{is\text{-}W\text{-}hinit}_{\mathcal{U}_k}(\mathcal{X}) :\equiv \Pi_{(\mathcal{Y} : \mathsf{W\text{-}Alg}_{\mathcal{U}_k}(A,B))} \mathsf{is\text{-}contr}(\mathsf{W\text{-}Hom}\ \mathcal{X}\ \mathcal{Y})$$

Definition 4.21. For $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, $\mathcal{X}$: $\mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A, B)$ there is a designated *identity* homomorphism from $\mathcal{X}$ to itself, defined by

$$\mathsf{W\text{-}Id\text{-}Hom}\ (C, c) :\equiv (\mathsf{id}_C; a, t \mapsto \mathsf{refl}_{c(a,t)})$$

As before, we denote this homomorphism by $1_{\mathcal{X}}$.

Definition 4.22. For $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, $\mathcal{X}$: $\mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A, B)$, $\mathcal{Y}$: $\mathsf{W\text{-}Alg}_{\mathcal{U}_k}(A, B)$, $\mathcal{Z}$: $\mathsf{W\text{-}Alg}_{\mathcal{U}_l}(A, B)$ and homomorphisms μ : $\mathsf{W\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}$, ν : $\mathsf{W\text{-}Hom}\ \mathcal{Y}\ \mathcal{Z}$, the *composition* of μ and ν is a homomorphism from $\mathcal{X}$ to $\mathcal{Z}$ defined by

$$\mathsf{W\text{-}Comp\text{-}Hom}\ (C, c)\ (D, d)\ (E, e)\ (f, \gamma)\ (g, \delta) :\equiv \Big(g \circ f; a, t \mapsto \mathsf{ap}_g(\gamma(a, t)) \cdot \delta(a, f \circ t)\Big)$$

As before, we often leave out the first three arguments and denote the composition by $\nu \circ \mu$.

Definition 4.23. For $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, $\mathcal{X}$: $\mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A, B)$, $\mathcal{Y}$: $\mathsf{W\text{-}Alg}_{\mathcal{U}_k}(A, B)$ we define the type of *isomorphisms* between $\mathcal{X}$ and $\mathcal{Y}$ as

$$\mathsf{W\text{-}Alg\text{-}Iso}\ \mathcal{X}\ \mathcal{Y} :\equiv \Sigma_{(\rho : \mathsf{W\text{-}Hom}\ \mathcal{X}\ \mathcal{Y})} \Big(\Sigma_{(\mu : \mathsf{W\text{-}Hom}\ \mathcal{Y}\ \mathcal{X})} \mu \circ \rho = 1_{\mathcal{X}}\Big) \times \Big(\Sigma_{(\nu : \mathsf{W\text{-}Hom}\ \mathcal{Y}\ \mathcal{X})} \rho \circ \nu = 1_{\mathcal{Y}}\Big)$$

LEMMA 4.24. *For $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, $\mathcal{X}$: $\mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A, B)$, $\mathcal{Y}$: $\mathsf{W\text{-}Fib\text{-}Alg}_{\mathcal{U}_k}(A, B)\ \mathcal{X}$ and μ, ν : $\mathsf{W\text{-}Fib\text{-}Hom}\ \mathcal{X}\ \mathcal{Y}$, the path space $\mu = \nu$ is equivalent to the space of fibered 2-cells between μ and ν:*

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

PROOF. Let algebras (C, c) : $\mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A, B)$, (E, e) : $\mathsf{W\text{-}Fib\text{-}Alg}_{\mathcal{U}_k}(A, B)$ (C, c) and homomorphisms $(f, \gamma), (g, \delta)$: $\mathsf{W\text{-}Fib\text{-}Hom}$ (C, c) (E, e) be given. We have

$$\begin{aligned}
&(f, \gamma) = (g, \delta) && \simeq \\
&\Sigma_{\alpha : f = g}\Big(\delta = \mathsf{tr}_{\left(h \mapsto \Pi_a \Pi_t h(c(a,t)) = e(a,t,\lambda_b h(t\ b))\right)}(\alpha)\ \gamma\Big) && \simeq \\
&\Sigma_{\alpha : f = g}\Big(\delta = \lambda_a \lambda_t {}^{=}\mathsf{Eq}^{\Pi}(\alpha, c(a,t))^{-1} \cdot \gamma(a,t) \cdot \mathsf{ap}_{e(a,t)}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b {}^{=}\mathsf{Eq}^{\Pi}(\alpha, t\ b)))\Big) && \simeq \\
&\Sigma_{\alpha : f \sim g}\Big(\delta = \lambda_a \lambda_t\ \alpha(c(a,t))^{-1} \cdot \gamma(a,t) \cdot \mathsf{ap}_{e(a,t)}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b\ \alpha(t\ b)))\Big) && \simeq \\
&\Sigma_{\alpha : f \sim g}\Pi_a \Pi_t\Big(\delta(a,t) = \alpha(c(a,t))^{-1} \cdot \gamma(a,t) \cdot \mathsf{ap}_{e(a,t)}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b\ \alpha(t\ b)))\Big) && \simeq \\
&\Sigma_{\alpha : f \sim g}\Pi_a \Pi_t\Big(\alpha(c(a,t)) = \gamma(a,t) \cdot \mathsf{ap}_{e(a,t)}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b\ \alpha(t\ b))) \cdot \delta(a,t)^{-1}\Big) && \equiv \\
&\mathsf{W\text{-}Fib\text{-}Cell}\ (f, \gamma)\ (g, \delta)
\end{aligned}$$

□

COROLLARY 4.25. *For $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, $\mathcal{X}$: $\mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A, B)$, $\mathcal{Y}$: $\mathsf{W\text{-}Alg}_{\mathcal{U}_k}(A, B)$ and μ, ν : $\mathsf{W\text{-}Hom}$ $\mathcal{X}$ $\mathcal{Y}$, the path space $\mu = \nu$ is equivalent to the space of 2-cells between μ and ν:*

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

LEMMA 4.26. *Given $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$ and $\mathcal{X}, \mathcal{Y}$: $\mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A, B)$, the path space $\mathcal{X} = \mathcal{Y}$ is equivalent to the space of isomorphisms between $\mathcal{X}$ and $\mathcal{Y}$:*

$$\mathcal{X} = \mathcal{Y} \quad \simeq \quad \mathsf{W\text{-}Alg\text{-}Iso}\ \mathcal{X}\ \mathcal{Y}$$

PROOF. Let algebras $(C, c), (D, d)$: $\mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A, B)$ be given. We have

$$\begin{aligned}
&\mathsf{W\text{-}Alg\text{-}Iso}\ (C, c)\ (D, d) && \equiv \\
&\Sigma_{\rho : \left(\Sigma_{f : C \to D} \Pi_a \Pi_t f(c(a,t)) = d(a, f \circ t)\right)}\Big(\Sigma_{\mu : \left(\Sigma_{g : D \to C} \Pi_a \Pi_s g(d(a,s)) = c(a, g \circ s)\right)} \mu \circ \rho = 1_{(C,c)}\Big) \times \\
&\qquad\qquad\Big(\Sigma_{\nu : \left(\Sigma_{h : D \to C} \Pi_a \Pi_s h(d(a,s)) = c(a, h \circ s)\right)} \rho \circ \nu = 1_{(D,d)}\Big) && \simeq \\
&\Sigma_{f : C \to D} \Sigma_{\epsilon : (\Pi_a \Pi_t f(c(a,t)) = d(a, f \circ t))}\Big(\Sigma_{g : D \to C} \Sigma_{\gamma : (\Pi_a \Pi_s g(d(a,s)) = c(a, g \circ s))} P_{f,\epsilon,g,\gamma}\Big) \times \\
&\qquad\qquad\Big(\Sigma_{h : D \to C} \Sigma_{\delta : (\Pi_a \Pi_s h(d(a,s)) = c(a, h \circ s))} Q_{f,\epsilon,h,\delta}\Big)
\end{aligned}$$

where

$$\begin{aligned}
P_{f,\epsilon,g,\gamma} &:= \Big(g \circ f; a, t \mapsto \mathsf{ap}_g(\epsilon(a,t)) \cdot \gamma(a, f \circ t)\Big) = \Big(\mathsf{id}_C; a, t \mapsto \mathsf{refl}_{c(a,t)}\Big) \\
Q_{f,\epsilon,h,\delta} &:= \Big(f \circ h; a, s \mapsto \mathsf{ap}_f(\delta(a,s)) \cdot \epsilon(a, h \circ t)\Big) = \Big(\mathsf{id}_D; a, s \mapsto \mathsf{refl}_{d(a,s)}\Big)
\end{aligned}$$

Now we have

$$
\begin{aligned}
&P_{f,\epsilon,g,\gamma} && \equiv\\
&\Big(g \circ f; a, t \mapsto \mathsf{ap}_g(\epsilon(a,t)) \bullet \gamma(a, f \circ t)\Big) = \Big(\mathsf{id}_C; a, t \mapsto \mathsf{refl}_{c(a,t)}\Big) && \simeq\\
&\Sigma_{\alpha:g\circ f=\mathsf{id}_C}\Big(\lambda_a\lambda_t\mathsf{ap}_g(\epsilon(a,t)) \bullet \gamma(a, f\circ t)\Big) = \mathsf{tr}^{-1}_{\left(i\mapsto \Pi_a\Pi_t i(c(a,t))=c(a,i\circ t)\right)}(\alpha)\ \Big(\lambda_a\lambda_t\mathsf{refl}_{c(a,t)}\Big) && \simeq\\
&\Sigma_{\alpha:g\circ f=\mathsf{id}_C}\Big(\lambda_a\lambda_t\mathsf{ap}_g(\epsilon(a,t)) \bullet \gamma(a, f\circ t)\Big) = \lambda_a\lambda_t\\
&\qquad {}^{=}\mathsf{Eq}^{\Pi}(\alpha, c(a,t)) \bullet \mathsf{refl}_{c(a,t)} \bullet \Big(\mathsf{ap}_{c(a)}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b {}^{=}\mathsf{Eq}^{\Pi}(\alpha, t\ b)))\Big)^{-1} && \simeq\\
&\Sigma_{\alpha:g\circ f=\mathsf{id}_C}\Pi_a\Pi_t\Big(\mathsf{ap}_g(\epsilon(a,t)) \bullet \gamma(a, f\circ t)\Big) =\\
&\qquad {}^{=}\mathsf{Eq}^{\Pi}(\alpha, c(a,t)) \bullet \mathsf{refl}_{c(a,t)} \bullet \Big(\mathsf{ap}_{c(a)}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b {}^{=}\mathsf{Eq}^{\Pi}(\alpha, t\ b)))\Big)^{-1} && \simeq\\
&\Sigma_{\alpha:g\circ f\sim\mathsf{id}_C}\Pi_a\Pi_t\Big(\mathsf{ap}_g(\epsilon(a,t)) \bullet \gamma(a, f\circ t)\Big) =\\
&\qquad \alpha(c(a,t)) \bullet \mathsf{refl}_{c(a,t)} \bullet \Big(\mathsf{ap}_{c(a)}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b\ \alpha(t\ b)))\Big)^{-1} && \simeq\\
&\Sigma_{\alpha:g\circ f\sim\mathsf{id}_C}\Pi_a\Pi_t\Big(\alpha(c(a,t)) = \mathsf{ap}_g(\epsilon(a,t)) \bullet \gamma(a, f\circ t) \bullet \mathsf{ap}_{c(a)}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b\ \alpha(t\ b)))\Big)
\end{aligned}
$$

and analogously

$$
Q_{f,\epsilon,h,\delta} \simeq \Sigma_{\beta:f\circ h\sim\mathsf{id}_D}\Pi_a\Pi_s\Big(\beta(d(a,s)) = \mathsf{ap}_f(\delta(a,s)) \bullet \epsilon(a, h\circ s) \bullet \mathsf{ap}_{d(a)}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b\ \beta(s\ b)))\Big)
$$

Thus we can express $\mathsf{W\text{-}Alg\text{-}Iso}\ (C,c)\ (D,d)$ equivalently as the type

$$
\Sigma_{f:C\to D}\Sigma_{\epsilon:(\Pi_a\Pi_t f(c(a,t))=d(a,f\circ t))}\Sigma_{g:D\to C}\Sigma_{\alpha:g\circ f\sim\mathsf{id}_C}\Sigma_{h:D\to C}\Sigma_{\beta:f\circ h\sim\mathsf{id}_D}R_{f,\epsilon,g,\alpha}\times S_{f,\epsilon,h,\beta}
$$

where

$$
\begin{aligned}
R_{f,\epsilon,g,\alpha} &:\equiv \Sigma_{\gamma:(\Pi_a\Pi_s g(d(a,s))=c(a,g\circ s))}\Pi_a\Pi_t\\
&\qquad \alpha(c(a,t)) = \mathsf{ap}_g(\epsilon(a,t)) \bullet \gamma(a, f\circ t) \bullet \mathsf{ap}_{c(a)}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b\ \alpha(t\ b)))\\
S_{f,\epsilon,h,\beta} &:\equiv \Sigma_{\delta:(\Pi_a\Pi_s h(d(a,s))=c(a,h\circ s))}\Pi_a\Pi_s\\
&\qquad \beta(d(a,s)) = \mathsf{ap}_f(\delta(a,s)) \bullet \epsilon(a, h\circ s) \bullet \mathsf{ap}_{d(a)}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b\ \beta(s\ b)))
\end{aligned}
$$

Now we have

$$
\begin{aligned}
&S_{f,\epsilon,h,\beta} && \equiv\\
&\Sigma_{\delta:(\Pi_a\Pi_s h(d(a,s))=c(a,h\circ s))}\Pi_a\Pi_s\\
&\qquad \beta(d(a,s)) = \mathsf{ap}_f(\delta(a,s)) \bullet \epsilon(a, h\circ s) \bullet \mathsf{ap}_{d(a)}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b\ \beta(s\ b))) && \simeq\\
&\Pi_a\Pi_s\Sigma_{\delta:h(d(a,s))=c(a,h\circ s)}\\
&\qquad \beta(d(a,s)) = \mathsf{ap}_f(\delta) \bullet \epsilon(a, h\circ s) \bullet \mathsf{ap}_{d(a)}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b\ \beta(s\ b))) && \simeq\\
&\Pi_a\Pi_s\Sigma_{\delta:h(d(a,s))=c(a,h\circ s)}\\
&\qquad \mathsf{ap}_f(\delta) = \beta(d(a,s)) \bullet \Big(\mathsf{ap}_{d(a)}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b\ \beta(s\ b)))\Big)^{-1} \bullet \epsilon(a, h\circ s)^{-1} && \simeq\\
&\Pi_a\Pi_s 1 && \simeq\\
&1
\end{aligned}
$$

by Lem. ??, ??, and the fact that g, α, h, β make f into an equivalence. This last fact also implies that the mapping $t \mapsto f \circ t$ is an equivalence between $B(a) \to C$ and $B(a) \to D$ by Lem. ??[8]. Thus the mapping $\gamma \mapsto \lambda_a \lambda_t \gamma(a, f \circ t)$ is an equivalence between the types $\Pi_{a:A}\Pi_{s:B(a)\to D} g(d(a,s)) = c(a, g \circ s)$ and $\Pi_{a:A}\Pi_{t:B(a)\to C} g(d(a, f\circ t)) = c(a, g \circ f \circ t)$. Thus: EdN:8

$$
\begin{aligned}
& R_{f,\epsilon,g,\alpha} && \equiv \\
& \Sigma_{\gamma:\left(\Pi_a\Pi_s g(d(a,s))=c(a,g\circ s)\right)}\Pi_a\Pi_t \\
& \qquad \alpha(c(a,t)) = \mathsf{ap}_g(\epsilon(a,t)) \bullet \gamma(a, f\circ t) \bullet \mathsf{ap}_{c(a)}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b\ \alpha(t\ b))) && \simeq \\
& \Sigma_{\gamma:\left(\Pi_a\Pi_t g(d(a,f\circ t))=c(a,g\circ f\circ t)\right)}\Pi_a\Pi_t \\
& \qquad \alpha(c(a,t)) = \mathsf{ap}_g(\epsilon(a,t)) \bullet \gamma(a,t) \bullet \mathsf{ap}_{c(a)}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b\ \alpha(t\ b))) && \simeq \\
& \Pi_a\Pi_t\Sigma_{\gamma:g(d(a,f\circ t))=c(a,g\circ f\circ t)} \\
& \qquad \alpha(c(a,t)) = \mathsf{ap}_g(\epsilon(a,t)) \bullet \gamma \bullet \mathsf{ap}_{c(a)}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b\ \alpha(t\ b))) && \simeq \\
& \Pi_a\Pi_t\Sigma_{\gamma:g(d(a,f\circ t))=c(a,g\circ f\circ t)} \\
& \qquad \gamma = \Big(\mathsf{ap}_g(\epsilon(a,t))\Big)^{-1} \bullet \alpha(c(a,t)) \bullet \Big(\mathsf{ap}_{c(a)}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b\ \alpha(t\ b)))\Big)^{-1} && \simeq \\
& \Pi_a\Pi_t 1 && \simeq \\
& 1
\end{aligned}
$$

by Lem. ??. Thus, we have

$$
\begin{aligned}
& \mathsf{W\text{-}Alg\text{-}Iso}\ (C,c)\ (D,d) && \simeq \\
& \Sigma_{f:C\to D}\Sigma_{g:D\to C}\Sigma_{\alpha:g\circ f\sim\mathsf{id}_C}\Sigma_{h:D\to C}\Sigma_{\beta:f\circ h\sim\mathsf{id}_D}\Pi_a\Pi_t f(c(a,t)) = d(a, f\circ t) && \simeq \\
& \Sigma_{f:C\to D}\mathsf{is\text{-}equiv}(f) \times \Pi_a\Pi_t f(c(a,t)) = d(a, f\circ t) && \simeq \\
& \Sigma_{p:C\simeq D}\Pi_a\Pi_t\big(\mathsf{fst}(p)\ c(a,t)\big) = d(a, \mathsf{fst}(p)\circ t) && \simeq \\
& \Sigma_{p:C=D}\Pi_a\Pi_t\Big(\mathsf{fst}({}^{=}\mathsf{Eq}^{\simeq}(p))\ c(a,t)\Big) = d\Big(a, \mathsf{fst}({}^{=}\mathsf{Eq}^{\simeq}(p))\circ t\Big) && \equiv \\
& \Sigma_{p:C=D}\Pi_a\Pi_t\Big(\mathsf{tr}_{X\mapsto X}(p)\ c(a,t) = d\Big(a, \mathsf{fst}({}^{=}\mathsf{Eq}^{\simeq}(p))\circ t\Big)\Big) && \simeq \\
& \Sigma_{p:C=D}\Pi_a\Pi_t\Big(c(a,t) = \mathsf{tr}^{-1}_{X\mapsto X}(p)\ d\Big(a, \mathsf{fst}({}^{=}\mathsf{Eq}^{\simeq}(p))\circ t\Big)\Big) && \simeq \\
& \Sigma_{p:C=D}\Big(c = \lambda_a\lambda_t\mathsf{tr}^{-1}_{X\mapsto X}(p)\ d\Big(a, \mathsf{fst}({}^{=}\mathsf{Eq}^{\simeq}(p))\circ t\Big)\Big) && \simeq \\
& \Sigma_{p:C=D}\Big(c = \mathsf{tr}^{-1}_{\left(X\mapsto\Pi_{a:A}(B(a)\to X)\to X\right)}(p)\ d\Big) && \simeq \\
& (C,c) = (D,d)
\end{aligned}
$$

□

As desired, the analogues of the lemmas from section 3.2 still hold in the setting of homotopy type theory:

LEMMA 4.27. *In $\mathcal{H}$, for $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, if an algebra $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A,B)$ satisfies the induction principle on the universe $\mathcal{U}_k$, it also satisfies the induction uniqueness principle on $\mathcal{U}_k$. In other words, we have*

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

[8]EDNOTE: Which says that if $f : B \to C$ is an equivalence, then so is the mapping $h \mapsto f \circ h$ between $A \to B$ and $A \to C$

PROOF. Let an algebra $(C, c) : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A, B)$ be given. To prove the induction uniqueness principle, take any algebra $(E, e) : \mathsf{W\text{-}Fib\text{-}Alg}_{\mathcal{U}_k}(A, B)\ (C, c)$ and homomorphisms $(f, \gamma), (g, \delta) : \mathsf{W\text{-}Fib\text{-}Hom}\ (C, c)\ (E, e)$. By Lem. 4.24, to show $(f, \gamma) = (g, \delta)$ it suffices to exhibit a fibered 2-cell between (f, γ) and (g, δ). For this we use the induction principle with the fibered algebra $\Big(f \sim g; x, p, r \mapsto \gamma(x, p) \cdot \mathsf{ap}_{e(x,p)}({}^{\Pi}\mathsf{Eq}^{=}(r)) \cdot \delta(x, p)^{-1}\Big)$ and we are done. $\square$

COROLLARY 4.28. *In $\mathcal{H}$, for $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, if an algebra $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A, B)$ satisfies the induction principle on the universe $\mathcal{U}_k$, it is homotopy-initial on $\mathcal{U}_k$. In other words, we have*

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

LEMMA 4.29. *In $\mathcal{H}$, for $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, if an algebra $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A, B)$ satisfies the recursion and recursion uniqueness principles on the universe $\mathcal{U}_k$ and $k \geq j$, then it satisfies the induction principle on $\mathcal{U}_k$. In other words, we have*

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

provided $k \geq j$.

PROOF. Let algebras $(C, c) : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A, B)$ and $(E, e) : \mathsf{W\text{-}Fib\text{-}Alg}_{\mathcal{U}_k}\ (C, c)$ be given. We use the recursion principle with the algebra $(\Sigma_{x:C} E(x); a, s \mapsto d(a, s))$ where

$$d(a, s) :\equiv \Big(c\big(a, \lambda_{b:B(a)} \mathsf{fst}(s\ b)\big), e\big(a, \lambda_{b:B(a)} \mathsf{fst}(s\ b), \lambda_{b:B(a)} \mathsf{snd}(s\ b)\big)\Big)$$

We note that the carrier type belongs to $\mathcal{U}_k$ as $j \leq k$. This gives us a homomorphism (u, θ), where $u : C \to \Sigma_{x:C} E(x)$ and $\theta : \Pi_{a:A} \Pi_{t:B(a) \to C}\, u(c(a, t)) = d(a, \lambda_{b:B(a)} u(t\ b))$. We can now form two homomorphisms

$$\Big(\mathsf{fst} \circ u; a, t \mapsto \mathsf{fst}({}^{=}\mathsf{Eq}^{\Sigma}(\theta(a, t)))\Big), \Big(\mathsf{id}_C; a, t \mapsto \mathsf{refl}_{c(a,t)}\Big) : \mathsf{W\text{-}Hom}\ (C, c)\ (C, c)$$

The recursion uniqueness principle tells us that these homomorphisms are equal. By Cor. 4.25 this means there exists a 2-cell (α, η) between them, i.e., we have

$$\alpha : \mathsf{fst} \circ u \sim \mathsf{id}_C$$
$$\eta : \Pi_{a:A} \Pi_{t:B(a) \to C} \Big(\alpha(c(a, t)) = \mathsf{fst}({}^{=}\mathsf{Eq}^{\Sigma}(\theta(a, t))) \cdot \mathsf{ap}_{c(a)}\big({}^{\Pi}\mathsf{Eq}^{=}(\lambda_{b:B(a)} \alpha(t\ b))\big) \cdot \mathsf{refl}_{c(a,t)}\Big)$$

Now it is easy to see that for any $a : A$, path $p : t_1 =_{B(a) \to C} t_2$, and $s : \Pi_{b:B(a)} E(t_1\ b)$, we have a higher path

$$\epsilon(a, p, s) : \mathsf{tr}_E\big(\mathsf{ap}_{c(a)}(p)\big)\ e(a, t_1, s) = e\Big(a, t_2, \big(\lambda_{b:B(a)} \mathsf{tr}_E({}^{=}\mathsf{Eq}^{\Pi}(p, b))\ (s\ b)\big)\Big)$$

To show this, we simply use path induction on p.

We can thus define the desired fibered homomorphism as

$$\Big(x \mapsto \mathsf{tr}_E(\alpha(x))\ \mathsf{snd}(u(x)); a, t \mapsto \phi(a, t)\Big) : \mathsf{W\text{-}Fib\text{-}Hom}\ (C, c)\ (E, e)$$

where $\phi(a, t)$ is the path

$$\begin{array}{c}
\mathsf{tr}_E\big(\alpha(c(a,t))\big)\ \mathsf{snd}(u(c(a,t))) \\
\Big|\ \text{via } \eta(a,t) \\
\mathsf{tr}_E\Big(\mathsf{fst}({}^{=}\mathsf{Eq}^{\Sigma}(\theta(a,t))) \cdot \mathsf{ap}_{c(a)}\big({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b\ \alpha(t\ b))\big) \cdot \mathsf{refl}_{c(a,t)}\Big)\ \mathsf{snd}(u(c(a,t))) \\
\Big| \\
\mathsf{tr}_E\Big(\mathsf{fst}({}^{=}\mathsf{Eq}^{\Sigma}(\theta(a,t))) \cdot \mathsf{ap}_{c(a)}\big({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b\ \alpha(t\ b))\big)\Big)\ \mathsf{snd}(u(c(a,t))) \\
\Big| \\
\mathsf{tr}_E\Big(\mathsf{ap}_{c(a)}\big({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b\ \alpha(t\ b))\big)\Big)\ \Big(\mathsf{tr}_E\big(\mathsf{fst}({}^{=}\mathsf{Eq}^{\Sigma}(\theta(a,t)))\big)\ \mathsf{snd}(u(c(a,t)))\Big) \\
\Big|\ \text{via } \mathsf{snd}({}^{=}\mathsf{Eq}^{\Sigma}(\theta(a,t))) \\
\mathsf{tr}_E\Big(\mathsf{ap}_{c(a)}\big({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b\ \alpha(t\ b))\big)\Big)\ e\big(a, \lambda_b\ \mathsf{fst}(u(t\ b)), \lambda_b\ \mathsf{snd}(u(t\ b))\big) \\
\Big|\ \epsilon\Big(a, {}^{\Pi}\mathsf{Eq}^{=}(\lambda_b\ \alpha(t\ b)), \lambda_b\ \mathsf{snd}(u(t\ b))\Big) \\
e\Big(a, t, \lambda_b \mathsf{tr}_E\Big({}^{=}\mathsf{Eq}^{\Pi}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_b\ \alpha(t\ b)), b)\Big)\ \mathsf{snd}(u(t\ b))\Big) \\
\Big| \\
e\Big(a, t, \lambda_b\ \mathsf{tr}_E(\alpha(t\ b))\ \mathsf{snd}(u(t\ b))\Big)
\end{array}$$

□

COROLLARY 4.30. *In $\mathcal{H}$, for $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, the following conditions on an algebra $\mathcal{X} : \mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_j}(A,B)$ are equivalent:*

(1) *$\mathcal{X}$ satisfies the induction principle on the universe $\mathcal{U}_k$*
(2) *$\mathcal{X}$ satisfies the recursion and recursion uniqueness principles on the universe $\mathcal{U}_k$*
(3) *$\mathcal{X}$ is homotopy-initial on the universe $\mathcal{U}_k$*

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

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

provided $j \geq i$. Furthermore, all 3 conditions are mere propositions.

PROOF. Conditions (1) and (3) are mere propositions by Lem. **??** (for the latter), Lem. **??**, 4.27 (for the former), and the fact that a family of mere propositions is itself a mere proposition. □

As in the case of the type 2, we have the following corollary:

COROLLARY 4.31. *In $\mathcal{H}$, for $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, two algebras $\mathcal{X}, \mathcal{Y} : \mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_j}(A,B)$ which are homotopy-initial on $\mathcal{U}_j$ are equal:*

$$\mathsf{is\text{-}W\text{-}hinit}_{\mathcal{U}_j}(\mathcal{X}) \to \mathsf{is\text{-}W\text{-}hinit}_{\mathcal{U}_j}(\mathcal{Y}) \to \mathcal{X} = \mathcal{Y}$$

PROOF. By Lem. 4.26 it suffices to construct an isomorphism between $\mathcal{X}$ and $\mathcal{Y}$. Since $\mathcal{X}$ and $\mathcal{Y}$ are both homotopy-initial on $\mathcal{U}_j$, there exist homomorphisms $\mu : \mathsf{W}\text{-}\mathsf{Hom}\ \mathcal{X}\ \mathcal{Y}$ and $\nu : \mathsf{W}\text{-}\mathsf{Hom}\ \mathcal{Y}\ \mathcal{X}$. Again by homotopy-initiality, we have $\nu \circ \mu = 1_{\mathcal{X}}$ and $\mu \circ \nu = 1_{\mathcal{Y}}$, which gives us the desired isomorphism. □

We can thus characterize W-types using the universal property of initiality as follows.

COROLLARY 4.32. *In $\mathcal{H}$ with* W*-types, for any $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, the algebra*

$$\left(\mathsf{W}^{\mathcal{U}_i}_{x:A} B(x), \lambda_a \lambda_t \mathsf{sup}^{A,x.B(x)}_{\mathcal{U}_i}(a,t)\right) : \mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_i}(A, B)$$

is homotopy-initial on any universe $\mathcal{U}_j$.

COROLLARY 4.33. *In $\mathcal{H}$ extended with an algebra $\mathcal{X}_{\mathcal{U}_i}(A, B) : \mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_i}(A, B)$ for any $\mathcal{U}_i$, $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, which is homotopy-initial on any universe $\mathcal{U}_j$, Martin-Löf's* W*-types are definable.*

PROOF. For any i, the algebra $A : \mathcal{U}_i, B : A \to \mathcal{U}_i \vdash \mathcal{X}_{\mathcal{U}_i}(A, B) : \mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_i}(A, B)$ is such that for any j, there exists a term $A : \mathcal{U}_i, B : A \to \mathcal{U}_i \vdash h^i_j : \mathsf{is}\text{-}\mathsf{W}\text{-}\mathsf{hinit}_{\mathcal{U}_j}(\mathcal{X}_{\mathcal{U}_i}(A, B))$. By Cor. 4.30, for any $j \geq i$ there is a term $A : \mathcal{U}_i, B : A \to \mathcal{U}_i \vdash r^i_j : \mathsf{has}\text{-}\mathsf{W}\text{-}\mathsf{ind}_{\mathcal{U}_j}(\mathcal{X}_{\mathcal{U}_i}(A, B))$. Since universes are cumulative, this implies that such a term r^i_j exists *for any* j. This in turn implies that W-types are definable. □

We have the following analogue of Lambek's lemma for W-types in homotopy type theory:

LEMMA 4.34. *Over $\mathcal{H}$, for $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, if an algebra $(C, c) : \mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_j}(A, B)$ is homotopy-initial on $\mathcal{U}_j$ and $j \geq i$, then the map from $\Sigma_{x:A} B(x) \to C$ to C given by c is an equivalence.*

PROOF. By abuse of notation we refer to both the curried and uncurried versions of the structure map by c. Since (C, c) is homotopy-initial on $\mathcal{U}_j$, it satisfies the recursion principle on $\mathcal{U}_j$. We use it with the algebra

$$\left(\Sigma_{x:A} B(x) \to C; a, s \mapsto \left(a, \lambda_{b:B(a)} c(s\ b)\right)\right)$$

We note that the carrier type belongs to $\mathcal{U}_j$ as $i \leq j$. This gives us a homomorphism (u, θ) where $u : C \to \left(\Sigma_{x:A} B(x) \to C\right)$ and $\theta : \Pi_a \Pi_t u(c(a,t)) = \left(a, \lambda_{b:B(a)} c(u(t\ b))\right)$. We can now form two homomorphisms

$$\left(c \circ u; a, t \mapsto \mathsf{ap}_c(\theta(a,t))\right), \left(\mathsf{id}_C; a, t \mapsto \mathsf{refl}_{c(a,t)}\right) : \mathsf{W}\text{-}\mathsf{Hom}\ (C, c)\ (C, c)$$

Since (C, c) is initial on $\mathcal{U}_j$, it satisfies the recursion uniqueness principle on $\mathcal{U}_j$, thus the above homomorphisms are equal. Thus, we have $c \circ u = \mathsf{id}_C$ and in particular there is a homotopy $\epsilon : c \circ u \sim \mathsf{id}_C$. For any a, t, the path $\theta(a,t) \cdot \mathsf{ap}_{(a,-)}({}^{\Pi}\mathsf{Eq}^{=}(\lambda_{b:B(a)} \epsilon(t\ b)))$ gives us $u(c(a,t)) = (a,t)$. Thus we have a homotopy $\eta : u \circ c \sim \mathsf{id}_{\Sigma_{x:A} B(x) \to C}$. □

COROLLARY 4.35. *Over $\mathcal{H}$, given $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$ and $a_1, a_2 : A$, $t_1 : B(a_1) \to \mathsf{W}$, $t_2 : B(a_2) \to \mathsf{W}$ we have*

$$\mathsf{sup}(a_1, t_1) = \mathsf{sup}(a_2, t_2) \quad \simeq \quad (a_1, t_1) = (a_2, t_2)$$

PROOF. By Lem. 4.34 and Cor. 4.32, the structure map sup defines an equivalence between $\Sigma_{x:A} B(x) \to \mathsf{W}$ and W. The rest follows from Lem. **??**. □

Finally, W-types with propositional computation rules still preserve homotopy levels, in the following sense:

LEMMA 4.36. *For $A : \mathcal{U}_i$, $B : A \to \mathcal{U}_i$, if A is an $(n+1)$-type, then so is $\mathsf{W}^{\mathcal{U}_i}_{x:A} B(x)$:*

$$\mathsf{is}\text{-}(\mathsf{n}+1)\text{-}\mathsf{type}(A) \to \mathsf{is}\text{-}(\mathsf{n}+1)\text{-}\mathsf{type}(\mathsf{W}^{\mathcal{U}_i}_{x:A} B(x))$$

PROOF. We need to show that for any $u, w : \mathsf{W}$ we have $\mathsf{is\text{-}n\text{-}type}(u = w)$. By induction on u, it suffices to show that for any a, t we have $\Pi_{w:\mathsf{W}} \mathsf{is\text{-}n\text{-}type}(\mathsf{sup}(a,t) = w)$ under the induction hypothesis $h : \Pi_{b:B(a)} \Pi_{w:\mathsf{W}} \mathsf{is\text{-}n\text{-}type}(t(b) = w)$.

By another induction, this time on w, it suffices to show that for any a', t' we have $\mathsf{is\text{-}n\text{-}type}\big(\mathsf{sup}(a,t) = \mathsf{sup}(a',t')\big)$ under the hypothesis $\Pi_{b:B(a')} \mathsf{is\text{-}n\text{-}type}\big(\mathsf{sup}(a,t) = t'(b)\big)$, which however is not necessary. Now we have

$$\begin{array}{ll}
\mathsf{sup}(a,t) = \mathsf{sup}(a',t') & \simeq \\
(a,t) = (a',t') & \simeq \\
\Sigma_{p:a=a'} \big(t = \mathsf{tr}_{x \mapsto B(x) \to \mathsf{W}}(p)\ t'\big) & \simeq \\
\Sigma_{p:a=a'} \big(t = \lambda_{b:B(a)} t'(\mathsf{tr}_B(p)\ b)\big) & \simeq \\
\Sigma_{p:a=a'} \Pi_{b:B(a)} \big(t(b) = t'(\mathsf{tr}_B(p)\ b)\big) &
\end{array}$$

where the first equality follows by Cor. 4.35. Since A is an $(n+1)$-type by assumption, we have $\mathsf{is\text{-}n\text{-}type}(a = a')$. For any p, b we have $\mathsf{is\text{-}n\text{-}type}\big(t(b) = t'(\mathsf{tr}_B(p)\ b)\big)$ by the hypothesis $h\big(b, t'(\mathsf{tr}_B(p)\ b)\big)$. Since a family of n-types is an n-type and a dependent sum of a family of n-types over an n-type is again an n-type, the last type in the above chain of equalities is an n-type. This finishes the proof. $\square$

We note that there is no restriction on the homotopy level of the fibers of B since they only appear contravariantly. Furthermore, we note that the lemma is no longer true if $n+1$ is replaced by n: if $A :\equiv 1$ and $B(x) :\equiv 1$, then $\mathsf{W}^{\mathcal{U}_0}_{x:A} B(x) \simeq 0$, which is of course not contractible.

4.3. Definability of inductive types

A development entirely analogous to the foregoing can be given for the type of natural numbers, lists, the second number class, and other inductive types. For those types which can be presented as a W-type, however - which includes the aforementioned examples - the characterization in terms of homotopy-initial algebras can be obtained as a corollary to the main theorem. We illustrate this for the type $\mathbb{N}$ of natural numbers, with the usual rules for zero and successor, the familiar induction principle, and the computation rules in *propositional* form. We refrain from giving these rules explicitly since they can be easily read off from the corresponding definitions of $\mathbb{N}$-algebras, $\mathbb{N}$-homomorphisms, etc.

Definition 4.37. Define the type of $\mathbb{N}$-*algebras* on a universe $\mathcal{U}_i$ as

$$\mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_i} :\equiv \Sigma_{C:\mathcal{U}_i} C \times (C \to C)$$

Definition 4.38. Define the type of *fibered* $\mathbb{N}$-*algebras* on a universe $\mathcal{U}_j$ over $\mathcal{X} : \mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ by

$$\mathbb{N}\text{-}\mathsf{Fib\text{-}Alg}_{\mathcal{U}_j}\ (C, c_0, c_s) :\equiv \Sigma_{E:C \to \mathcal{U}_j} E(c_0) \times \big(\Pi_{x:C} E(x) \to E(c_s\ x)\big)$$

Definition 4.39. Given algebras $\mathcal{X} : \mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ and $\mathcal{Y} : \mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_j}$, define the type of $\mathbb{N}$-*homomorphisms* from $\mathcal{X}$ to $\mathcal{Y}$ by

$$\mathbb{N}\text{-}\mathsf{Hom}\ (C, c_0, c_s)\ (D, d_0, d_s) :\equiv \Sigma_{f:C \to D} (f(c_0) = d_0) \times \big(\Pi_{x:C} f(c_s\ x) = d_s(f(x))\big)$$

Definition 4.40. Given algebras $\mathcal{X} : \mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ and $\mathcal{Y} : \mathbb{N}\text{-}\mathsf{Fib\text{-}Alg}_{\mathcal{U}_j}\ \mathcal{X}$, define the type of *fibered* $\mathbb{N}$-*homomorphisms* from $\mathcal{X}$ to $\mathcal{Y}$ by

$$\mathbb{N}\text{-}\mathsf{Fib\text{-}Hom}\ (C, c_0, c_s)\ (E, e_0, e_s) :\equiv \Sigma_{(f:\Pi_{x:C} E(x))} (f(c_0) = e_0) \times \big(\Pi_{x:C} f(c_s\ x) = d_s(x, f(x))\big)$$

Definition 4.41. An algebra $\mathcal{X} : \mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ *satisfies the recursion principle* on a universe $\mathcal{U}_j$ if for any algebra $\mathcal{Y} : \mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_j}$ there exists a $\mathbb{N}$-homomorphism between $\mathcal{X}$ and $\mathcal{Y}$:

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

Definition 4.42. An algebra $\mathcal{X} : \mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ *satisfies the induction principle* on a universe $\mathcal{U}_j$ if for any fibered algebra $\mathcal{Y} : \mathbb{N}\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_j}\ \mathcal{X}$ there exists a fibered $\mathbb{N}$-homomorphism between $\mathcal{X}$ and $\mathcal{Y}$:

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

Definition 4.43. An algebra $\mathcal{X} : \mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ satisfies the *recursion uniqueness principle* on a universe $\mathcal{U}_j$ if for any algebra $\mathcal{Y} : \mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_j}$ any two $\mathbb{N}$-homomorphisms between $\mathcal{X}$ and $\mathcal{Y}$ are equal:

$$\mathsf{has}\text{-}\mathbb{N}\text{-}\mathsf{rec}\text{-}\mathsf{uniq}_{\mathcal{U}_j}(\mathcal{X}) :\equiv \Pi_{(\mathcal{Y}:\mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_j})}\mathsf{is}\text{-}\mathsf{prop}(\mathbb{N}\text{-}\mathsf{Hom}\ \mathcal{X}\ \mathcal{Y})$$

Definition 4.44. An algebra $\mathcal{X} : \mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ satisfies the *induction uniqueness principle* on a universe $\mathcal{U}_j$ if for any fibered algebra $\mathcal{Y} : \mathbb{N}\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_j}\ \mathcal{X}$ any two fibered $\mathbb{N}$-homomorphisms between $\mathcal{X}$ and $\mathcal{Y}$ are equal:

$$\mathsf{has}\text{-}\mathbb{N}\text{-}\mathsf{ind}\text{-}\mathsf{uniq}_{\mathcal{U}_j}(\mathcal{X}) :\equiv \Pi_{(\mathcal{Y}:\mathbb{N}\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_j}\ \mathcal{X})}\mathsf{is}\text{-}\mathsf{prop}(\mathbb{N}\text{-}\mathsf{Fib}\text{-}\mathsf{Hom}\ \mathcal{X}\ \mathcal{Y})$$

Definition 4.45. An algebra $\mathcal{X} : \mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ is *homotopy-initial* on a universe $\mathcal{U}_j$ if for any algebra $\mathcal{Y} : \mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_j}$, the type of $\mathbb{N}$-homomorphisms between $\mathcal{X}$ and $\mathcal{Y}$ is contractible:

$$\mathsf{is}\text{-}\mathbb{N}\text{-}\mathsf{hinit}_{\mathcal{U}_j}(\mathcal{X}) :\equiv \Pi_{(\mathcal{Y}:\mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_j})}\mathsf{is}\text{-}\mathsf{contr}(\mathbb{N}\text{-}\mathsf{Hom}\ \mathcal{X}\ \mathcal{Y})$$

We can encode the type of natural numbers as a W-type with $A :\equiv \mathsf{2}$ and B given by $0 \mapsto 0, 1 \mapsto 1$. By the propositional computation rules for $\mathsf{2}$ we have $B(0) = 0$, $B(1) = 1$. Thus in particular we have an equivalence $F : B(0) \to \mathsf{0}$ (and also $G : B(1) \to \mathsf{1}$ but this one is redundant).

LEMMA 4.46. *There is an equivalence*

$$\mathsf{W}\text{-}\mathsf{to}\text{-}\mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_i} : \mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_i}(A, B) \to \mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$$

PROOF. Fix $C : \mathcal{U}_i$. We have an equivalence

$$\begin{array}{c} \Pi_{a:2}(B(a) \to C) \to C \\ \Big\downarrow c \mapsto (c(0), c(1)) \\ \Big((B(0) \to C) \to C\Big) \times \Big((B(1) \to C) \to C\Big) \end{array}$$

Since $B(0) = 0$, the type $B(0) \to C$ is contractible, with center $\lambda_{b:B(0)}\mathsf{0rec}(C, F\ b)$. We thus have an equivalence

$$\begin{array}{c} (B(0) \to C) \to C \\ \Big\downarrow u \mapsto u\big(\lambda_{b:B(0)}\mathsf{0rec}(C, F\ b)\big) \\ C \end{array}$$

Since $B(1) = 1$, the map $x \mapsto \lambda_{\_}\, x$ is an equivalence from C to $B(1) \to C$. Thus, we have an equivalence

$$\begin{array}{c}(B(1) \to C) \to C \\ \Big\downarrow u \mapsto \lambda_{x:C} u(\lambda_{\_}\, x) \\ C \to C\end{array}$$

Putting this all together, we see that the map

$$(C, c) \mapsto \Big(C, c\big(0, \lambda_b \mathsf{0rec}(C, F\ b)\big), \lambda_{x:C} c(1, \lambda_{\_}\, x)\Big)$$

is an equivalence from $\mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_i}(A, B)$ to $\mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$. $\square$

LEMMA 4.47. *For any* $\mathcal{X} : \mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_i}(A, B)$ *there is an equivalence*

$$\mathsf{W}\text{-}\mathsf{to}\text{-}\mathbb{N}\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_j} : \mathsf{W}\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_j}(A, B)\ \mathcal{X} \to \mathbb{N}\text{-}\mathsf{Fib}\text{-}\mathsf{Alg}_{\mathcal{U}_j}(\mathsf{W}\text{-}\mathsf{to}\text{-}\mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_i}(\mathcal{X}))$$

PROOF. Let an algebra $(C, c) : \mathsf{W}\text{-}\mathsf{Alg}_{\mathcal{U}_i}(A, B)$ be given and fix $E : C \to \mathcal{U}_j$. We have an equivalence

$$\begin{array}{c}\Pi_{a:2}\Pi_{t:B(a)\to C}(\Pi_{b:B(a)}E(t\ b)) \to E(c(a,t)) \\ \Big\downarrow e \mapsto (e(0), e(1)) \\ \Big(\Pi_{t:B(0)\to C}(\Pi_{b:B(0)}E(t\ b)) \to E(c(0,t))\Big) \times \Big(\Pi_{t:B(1)\to C}(\Pi_{b:B(1)}E(t\ b)) \to E(c(1,t))\Big)\end{array}$$

Since $B(0) = 0$, the type $B(0) \to C$ is contractible, with center $\lambda_{b:B(0)}\mathsf{0rec}(C, F\ b)$. Likewise, $\Pi_{b:B(0)}E(\mathsf{0rec}(C, F\ b))$ is contractible, with center $\lambda_{b:B(0)}\mathsf{0rec}\big(E(\mathsf{0rec}(C, F\ b)), F\ b\big)$. We thus have equivalences

$$\begin{array}{c}\Pi_{t:B(0)\to C}(\Pi_{b:B(0)}E(t\ b)) \to E(c(0,t)) \\ \Big\downarrow u \mapsto u\big(\lambda_{b:B(0)}\mathsf{0rec}(C, F\ b)\big) \\ \big(\Pi_{b:B(0)}E(\mathsf{0rec}(C, F\ b))\big) \to E(c(0, \lambda_{b:B(0)}\mathsf{0rec}(C, F\ b))) \\ \Big\downarrow u \mapsto u\big(\lambda_{b:B(0)}\mathsf{0rec}\big(E(\mathsf{0rec}(C, F\ b)), F\ b\big)\big) \\ E(c(0, \lambda_{b:B(0)}\mathsf{0rec}(C, F\ b)))\end{array}$$

Since $B(1) = 1$, the map $x \mapsto \lambda_{\_}\, x$ is an equivalence from C to $B(1) \to C$. Likewise, for any $x : C$ the map $y \mapsto \lambda_{\_}\, y$ is an equivalence from $E(x)$ to $B(1) \to E(x)$. Thus, we have equivalences

$$\begin{array}{c}\Pi_{t:B(1)\to C}(\Pi_{b:B(1)}E(t\ b)) \to E(c(1,t)) \\ \Big\downarrow u \mapsto \lambda_{x:C}u(\lambda_\_\ x) \\ \Pi_{x:C}(B(1) \to E(x)) \to E(c(1,\lambda_\_\ x)) \\ \Big\downarrow u \mapsto \lambda_{x:C}\lambda_{y:E(x)}u(x,\lambda_\_\ y) \\ \Pi_{x:C}E(x) \to E(c(1,\lambda_\_\ x))\end{array}$$

Putting this all together, we see that the map

$$(E,e) \mapsto \Big(E, e\big(0, \lambda_b \mathsf{0rec}(C, F\ b), \lambda_b \mathsf{0rec}(E(\mathsf{0rec}(C, F\ b)), F\ b)\big), \lambda_{x:C}\lambda_{y:E(x)}e(1,\lambda_\_\ x, \lambda_\_\ y)\Big)$$

is an equivalence from $\mathsf{W\text{-}Fib\text{-}Alg}_{\mathcal{U}_j}(A,B)\ (C,c)$ to $\mathbb{N}\text{-}\mathsf{Fib\text{-}Alg}_{\mathcal{U}_j}(\mathsf{W\text{-}to\text{-}}\mathbb{N}\mathsf{\text{-}Alg}_{\mathcal{U}_i}\ (C,c))$. □

LEMMA 4.48. *For any* $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_i}(A,B)$ *and* $\mathcal{Y} : \mathsf{W\text{-}Fib\text{-}Alg}_{\mathcal{U}_j}(A,B)\ \mathcal{X}$ *we have*

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

PROOF. Let algebras $(C,c) : \mathsf{W\text{-}Alg}_{\mathcal{U}_i}(A,B)$ and $(E,e) : \mathsf{W\text{-}Fib\text{-}Alg}_{\mathcal{U}_j}(A,B)\ (C,c)$ be given and fix $f : \Pi_{x:C}E(x)$. We have an equivalence

$$\begin{array}{c}\Pi_{a:2}\Pi_{t:B(a)\to C}f(c(a,t)) = e(a,t,\lambda_{b:B(a)}f(t\ b)) \\ \Big\downarrow \simeq \\ \Big(\Pi_{t:B(0)\to C}f(c(0,t)) = e(0,t,\lambda_b f(t\ b))\Big) \times \Big(\Pi_{t:B(1)\to C}f(c(1,t)) = e(1,t,\lambda_b f(t\ b))\Big)\end{array}$$

Since $B(0) = 0$, the type $B(0) \to C$ is contractible, with center $\lambda_{b:B(0)}\mathsf{0rec}(C, F\ b)$. Furthermore, since all functions out of $\mathsf{0}$ are equal, we have

$$\lambda_b f(\mathsf{0rec}(C, F\ b)) = \lambda_b \mathsf{0rec}(E(\mathsf{0rec}(C, F\ b)), F\ b)$$

This implies the following equivalences:

$$\begin{array}{c}\Pi_{t:B(0)\to C}f(c(0,t)) = e(0,t,\lambda_b f(t\ b)) \\ \Big| \simeq \\ f\big(c(0,\lambda_b \mathsf{0rec}(C,F\ b))\big) = e\big(0, \lambda_b \mathsf{0rec}(C,F\ b), \lambda_b f(\mathsf{0rec}(C,F\ b))\big) \\ \Big| \simeq \\ f\big(c(0,\lambda_b \mathsf{0rec}(C,F\ b))\big) = e\big(0, \lambda_b \mathsf{0rec}(C,F\ b), \lambda_b \mathsf{0rec}(E(\mathsf{0rec}(C,F\ b)), F\ b)\big)\end{array}$$

Since $B(1) = 1$, the map $x \mapsto \lambda_\_\ x$ is an equivalence from C to $B(1) \to C$. Thus, we have an equivalence

$$\begin{array}{c}\Pi_{t:B(1)\to C}f(c(1,t)) = e(1,t,\lambda_b f(t\ b)) \\ \Big| \simeq \\ \Pi_{x:C}f\big(c(1,\lambda_\_\ x)\big) = e\big(1,\lambda_\_\ x, \lambda_\_\ f(x)\big)\end{array}$$

This finishes the proof. □

COROLLARY 4.49. *For any* $\mathcal{X} : \mathsf{W\text{-}Alg}_{\mathcal{U}_i}(A, B)$ *and* $\mathcal{Y} : \mathsf{W\text{-}Alg}_{\mathcal{U}_j}(A, B)$ *we have*

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

COROLLARY 4.50. *For any* $\mathcal{X} : \mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ *we have*

$$\begin{aligned}
\mathsf{has\text{-}}\mathbb{N}\text{-}\mathsf{rec}_{\mathcal{U}_j}(\mathcal{X}) &\simeq \mathsf{has\text{-}W\text{-}rec}_{\mathcal{U}_j}(\mathsf{W\text{-}to\text{-}}\mathbb{N}\text{-}\mathsf{Alg}^{-1}_{\mathcal{U}_i}(\mathcal{X}))\\
\mathsf{has\text{-}}\mathbb{N}\text{-}\mathsf{ind}_{\mathcal{U}_j}(\mathcal{X}) &\simeq \mathsf{has\text{-}W\text{-}ind}_{\mathcal{U}_j}(\mathsf{W\text{-}to\text{-}}\mathbb{N}\text{-}\mathsf{Alg}^{-1}_{\mathcal{U}_i}(\mathcal{X}))\\
\mathsf{has\text{-}}\mathbb{N}\text{-}\mathsf{rec\text{-}uniq}_{\mathcal{U}_j}(\mathcal{X}) &\simeq \mathsf{has\text{-}W\text{-}rec\text{-}uniq}_{\mathcal{U}_j}(\mathsf{W\text{-}to\text{-}}\mathbb{N}\text{-}\mathsf{Alg}^{-1}_{\mathcal{U}_i}(\mathcal{X}))\\
\mathsf{has\text{-}}\mathbb{N}\text{-}\mathsf{ind\text{-}uniq}_{\mathcal{U}_j}(\mathcal{X}) &\simeq \mathsf{has\text{-}W\text{-}ind\text{-}uniq}_{\mathcal{U}_j}(\mathsf{W\text{-}to\text{-}}\mathbb{N}\text{-}\mathsf{Alg}^{-1}_{\mathcal{U}_i}(\mathcal{X}))\\
\mathsf{is\text{-}}\mathbb{N}\text{-}\mathsf{hinit}_{\mathcal{U}_j}(\mathcal{X}) &\simeq \mathsf{is\text{-}W\text{-}hinit}_{\mathcal{U}_j}(\mathsf{W\text{-}to\text{-}}\mathbb{N}\text{-}\mathsf{Alg}^{-1}_{\mathcal{U}_i}(\mathcal{X}))
\end{aligned}$$

COROLLARY 4.51. *In* $\mathcal{H}$*, the following conditions on an algebra* $\mathcal{X} : \mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_i}$ *are equivalent:*

(1) $\mathcal{X}$ *satisfies the induction principle on the universe* $\mathcal{U}_j$
(2) $\mathcal{X}$ *satisfies the recursion and recursion uniqueness principles on the universe* $\mathcal{U}_j$
(3) $\mathcal{X}$ *is homotopy-initial on the universe* $\mathcal{U}_j$

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

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

provided $j \geq i$*. Furthermore, all 3 conditions are mere propositions.*

We can thus characterize the type $\mathbb{N}$ using the universal property of initiality as follows.

COROLLARY 4.52. *In* $\mathcal{H}$ *with natural numbers, the algebra* $(\mathbb{N}, \mathsf{z}, \mathsf{suc}(-)) : \mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_0}$ *is homotopy-initial on any universe* $\mathcal{U}_j$*.*

COROLLARY 4.53. *In* $\mathcal{H}$ *extended with an algebra* $\mathcal{X} : \mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_0}$ *which is homotopy-initial on any universe* $\mathcal{U}_j$*, the type* $\mathbb{N}$ *with propositional computation rules is definable.*

PROOF. We have an algebra $\cdot \vdash \mathcal{X} : \mathbb{N}\text{-}\mathsf{Alg}_{\mathcal{U}_0}$ such that for any j, there exists a term $\cdot \vdash h_j : \mathsf{is\text{-}}\mathbb{N}\text{-}\mathsf{hinit}_{\mathcal{U}_j}(\mathcal{X})$. Since the requirement $j \geq 0$ always holds, Cor. 4.51 implies that for any j, we have a term $\cdot \vdash r_j : \mathsf{has\text{-}}\mathbb{N}\text{-}\mathsf{ind}_{\mathcal{U}_j}(\mathcal{X})$. This implies that the type $\mathbb{N}$ with propositional computation rules is definable. □

5. OLD PART

5.1. Extensional W-types

We briefly recall the theory of W-types in fully extensional type theories. Let us begin by recalling the rules for W-types from [Martin-Löf 1984]. To state them more conveniently, we sometimes write W instead of $(\mathsf{W}x : A)B(x)$.

— W-formation rule.

$$\frac{A : \mathsf{type} \qquad x : A \vdash B(x) : \mathsf{type}}{(\mathsf{W}x : A)B(x) : \mathsf{type}}$$

— W-introduction rule.

$$\frac{a : A \qquad t : B(a) \to W}{\mathsf{sup}(a, t) : W}$$

— W-elimination rule.

$$\frac{\begin{array}{l} w : W \vdash C(w) : \mathsf{type} \\ x : A\,, u : B(x) \to W\,, v : (\Pi y : B(x))C(u(y)) \vdash \\ \qquad c(x,u,v) : C(\mathsf{sup}(x,u)) \end{array}}{w : W \vdash \mathsf{wrec}(w,c) : C(w)}$$

— W-computation rule.

$$\frac{\begin{array}{l} w : W \vdash C(w) : \mathsf{type} \\ x : A\,, u : B(x) \to W\,, v : (\Pi y : B(x))C(u(y)) \vdash \\ \qquad c(x,u,v) : C(\mathsf{sup}(x,u)) \end{array}}{\begin{array}{l} x : A, u : B(x) \to W \vdash \mathsf{wrec}(\mathsf{sup}(x,u),c) = \\ \qquad c(x,u,\lambda y.\mathsf{wrec}(u(y),c)) : C(\mathsf{sup}(x,u))\,. \end{array}}$$

W-types can be seen informally as the free algebras for signatures with operations of possibly infinite arity, but no equations. Indeed, the premisses of the formation rule above can be thought of as specifying a signature that has the elements of A as operations and in which the arity of $a : A$ is the cardinality of the type $B(a)$. Then, the introduction rule specifies the canonical way of forming an element of the free algebra, and the elimination rule can be seen as the propositions-as-types translation of the appropriate induction principle.

In extensional type theories, this informal description can easily be turned into a precise mathematical characterization. To do so, let us use the theory $\mathcal{H}_{\mathrm{ext}}$ obtained by extending $\mathcal{H}$ with the reflection rule in (1). Let $\mathcal{C}(\mathcal{H}_{\mathrm{ext}})$ be the category with types as objects and elements $f : A \to B$ as maps, in which two maps are considered equal if and only if they are definitionally equal. The premisses of the introduction rule determines the *polynomial endofunctor* $P : \mathcal{C}(\mathcal{H}_{\mathrm{ext}}) \to \mathcal{C}(\mathcal{H}_{\mathrm{ext}})$ defined by

$$P(X) \coloneqq (\Sigma x : A)(B(x) \to X)\,.$$

A P-algebra is a pair consisting of a type C and a function $s_C : PC \to C$, called the structure map of the algebra. The formation rule gives us an object $W \coloneqq (\mathsf{W}x : A)B(x)$ and the introduction rule (in combination with the rules for Π-types and Σ-types) provides a structure map

$$s_W : PW \to W\,.$$

The elimination rule, on the other hand, states that in order for the projection $\pi_1 : C \to W$, where $C \coloneqq (\Sigma w : W)C(w)$, to have a section s, as in the diagram

$$\begin{tikzcd} & C \arrow[d, "\pi_1"] \\ W \arrow[ur, "s"] \arrow[r, "1_W"'] & W, \end{tikzcd}$$

it is sufficient for the type C to have a P-algebra structure over W. Finally, the computation rule states that the section s given by the elimination rule is also a P-algebra homomorphism.

The foregoing elimination rule implies what we call the *simple* W-elimination rule:

$$\frac{C : \mathsf{type} \qquad x : A, v : B(x) \to C \vdash c(x,v) : C}{w : W \vdash \mathsf{simp\text{-}wrec}(w,c) : C}$$

This can be recognized as a recursion principle for maps from W into P-algebras, since the premisses of the rule describe exactly a type C equipped with a structure map s_C :

$PC \to C$. For this special case of the elimination rule, the corresponding computation rule again states that the function

$$\lambda w.\mathsf{simp\text{-}wrec}(w, c) : W \to C \,,$$

where $c(x, v) = s_C(\mathsf{pair}(x, v))$ for $x : A$ and $v : B(x) \to C$, is a P-algebra homomorphism. Moreover, this homomorphism can then be shown to be definitionally unique using the elimination rule, the principle of function extensionality and the reflection rule. The converse implication also holds: one can derive the general W-elimination rule from the simple elimination rule and the following η-rule

$$\frac{\begin{array}{l} C : \mathsf{type} \qquad w : W \vdash h(w) : C \\ x : A, v : B(x) \to C \vdash c(x, v) : C \\ x : A\,, u : B(x) \to W \vdash h\,(\mathsf{sup}(x, u)) = c(x, \lambda y.hu(y)) : C \end{array}}{w : W \vdash h(w) = \mathsf{simp\text{-}wrec}(w, c) : C}$$

stating the uniqueness of the simp-wrec term among algebra maps. Overall, we therefore have that in $\mathcal{H}_{\text{ext}}$ induction and recursion are interderivable:

Induction	$\Leftrightarrow$	**Recursion**
W-elimination		Simple W-elimination
W-computation		Simple W-computation + η-rule

Finally, observe that what we are calling recursion is equivalent to the statement that the type W, equipped with the structure map $s_W : PW \to W$ is the initial P-algebra. Indeed, assume the simple elimination rule, the simple computation rule and the η-rule; then for any P-algebra $s_C : PC \to C$, there is a function $f : W \to C$ by the simple elimination rule, which is a homomorphism by the computational rule, and is the unique such homomorphism by the η-rule. The converse implication from initiality to recursion is just as direct. Thus, in the extensional theory, to have an initial algebra for the endofunctor P is the same thing as having a type W satisfying the introduction, elimination and computation rules above. Section 5.3 will be devoted to generalizing this equivalence to the setting of Homotopy Type Theory.

5.2. Inductive types as W-types

To conclude our review, recall that in extensional type theory, many inductive types can be reduced to W-types. We mention the following examples, among many others (see [Martin-Löf 1984], [Dybjer 1997], [Goguen and Luo 1993], [Moerdijk and Palmgren 2000], [Gambino and Hyland 2004], [Abbott et al. 2005]):

(1) *Natural numbers*. The usual rules for $\mathbb{N}$ as an inductive type can be derived from its formalization as the following W-type. Consider the signature with two operations, one of which has arity 0 and one of which has arity 1; it is presented type-theoretically by a dependent type with corresponding polynomial functor (naturally isomorphic to)

$$P(X) = 1 + X \,,$$

and the natural numbers $\mathbb{N}$ together with the canonical element $0 : \mathbb{N}$ and the successor function $s : \mathbb{N} \to \mathbb{N}$ form an initial P-algebra

$$(0, s) : 1 + \mathbb{N} \to \mathbb{N} \,.$$

(2) *Second number class.* As shown in [Martin-Löf 1984], the second number class can be obtained as a W-type determined by the polynomial functor

$$P(X) = 1 + X + (\mathbb{N} \to X) \,.$$

This has algebras with three operations, one of arity 0, one of arity 1, and one of arity (the cardinality of) $\mathbb{N}$.

5.3. Intensional W-types

We begin with an example which serves to illustrate, in an especially simple case, some aspects of our theory. The type of Boolean truth values is not a W-type, but it can be formulated as an inductive type in the familiar way by means of formation, introduction, elimination, and computation rules. It then has an "up to homotopy" universal property of the same general kind as the one that we shall formulate in section 5.5 below for W-types, albeit in a simpler form.

5.4. Preliminary example

The standard rules for the type 2 given in [Nordstrom et al. 2000, Section 5.1] can be stated equivalently as follows.

— 2-formation rule.

$$\mathsf{2} : \mathsf{type}\,.$$

— 2-introduction rules.

$$\mathsf{0} : \mathsf{2}\,, \qquad \mathsf{1} : \mathsf{2}\,.$$

— 2-elimination rule.

$$\frac{x : \mathsf{2} \vdash C(x) : \mathsf{type} \qquad c_0 : C(0) \qquad c_1 : C(1)}{x : \mathsf{2} \vdash \mathsf{2rec}_{\mathcal{U}_i}(x, c_0, c_1) : C(x)}$$

— 2-computation rules.

$$\frac{x : \mathsf{2} \vdash C(x) : \mathsf{type} \qquad c_0 : C(0) \qquad c_1 : C(1)}{\begin{cases} \mathsf{2rec}_{\mathcal{U}_i}(0, c_0, c_1) = c_0 : C(0)\,, \\ \mathsf{2rec}_{\mathcal{U}_i}(1, c_0, c_1) = c_1 : C(1)\,. \end{cases}}$$

Although these rules are natural ones to consider in the intensional setting, they do not imply a strict universal property. For example, given a type C and elements $c_0, c_1 : C$, the function $\lambda x.\mathsf{2rec}_{\mathcal{U}_i}(x, c_0, c_1) : \mathsf{2} \to C$ cannot be shown to be definitionally unique among the functions $f : \mathsf{2} \to C$ with the property that $f(0) = c_0 : C$ and $f(1) = c_1 : C$. The best that one can do by using 2-elimination over a suitable identity type, and function extensionality, is to show that it is unique among all such maps up to an identity term, which itself is unique up to a higher identity, which in turn is unique up to $\ldots$. This sort of weak ω-universality, which apparently involves infinitely much data, can nonetheless be captured directly within the system of type theory (without resorting to coinduction) using ideas from higher category theory. To do so, let us define a *2-algebra* to be a type C equipped with two elements $c_0\,, c_1 : C$. Then, a *weak homomorphism* of 2-algebras $(f, p_0, p_1) : (C, c_0, c_1) \to (D, d_0, d_1)$ consists of a function $f : C \to D$ together with identity terms

$$p_0 : \mathsf{Id}_D(f(c_0), d_0)\,, \qquad p_1 : \mathsf{Id}_D(f(c_1), d_1)\,.$$

This is a *strict homomorphism* when $f(c_0) = d_0 : D$, $f(c_1) = d_1 : D$ and the identity terms p_0 and p_1 are the corresponding reflexivity terms. We can then define the type of weak homomorphisms from (C, c_0, c_1) to (D, d_0, d_1) by letting

$$\mathsf{2\text{-}Alg}\big[(C, c_0, c_1), (D, d_0, d_1)\big] \coloneqq (\Sigma f : C \to D)\mathsf{Id}(f(c_0), d_0) \times \mathsf{Id}_D(f(c_1), d_1)\,.$$

The weak universality condition on the 2-algebra $(2, 0, 1)$ that we seek can now be determined as follows.

Definition 5.1. A 2-algebra (C, c_0, c_1) is *homotopy-initial* if for any 2-algebra (D, d_0, d_1), the type

$$\mathsf{2\text{-}Alg}\big[(C, c_0, c_1), (D, d_0, d_1)\big]$$

is contractible.

The notion of homotopy initiality, or h-initiality for short, captures in a precise way the informal idea that there is essentially one weak algebra homomorphism $(2, 0, 1) \to (C, c_0, c_1)$. Moreover, h-initiality can be shown to follow from the rules of inference for 2 stated above. Indeed, the computation rules for 2 stated above evidently make the function

$$\lambda x.\mathsf{2rec}_{\mathcal{U}_i}(x, c_0, c_1) : 2 \to C$$

into a *strict* algebra map, a stronger condition than is required for h-initiality. Relaxing these definitional equalities to propositional ones, we arrive at the following rules.

— Propositional 2-computation rules.

$$\frac{x : 2 \vdash C(x) : \mathsf{type} \qquad c_0 : C(0) \qquad c_1 : C(1)}{\begin{cases} \mathsf{2comp}_0(c_0, c_1) : \mathsf{Id}_{C(0)}\big(\mathsf{2rec}_{\mathcal{U}_i}(0, c_0, c_1), c_0\big)\,, \\ \mathsf{2comp}_1(c_0, c_1) : \mathsf{Id}_{C(1)}\big(\mathsf{2rec}_{\mathcal{U}_i}(1, c_0, c_1), c_1\big)\,. \end{cases}}$$

This variant is not only still sufficient for h-initiality, but also necessary, as we state precisely in the following.

PROPOSITION 5.2. *Over the type theory $\mathcal{H}$, the formation, introduction, elimination, and propositional computation rules for 2 are equivalent to the existence of a homotopy-initial 2-algebra.*

PROOF PROOF SKETCH. Suppose we have a type 2 satisfying the stated rules. Then clearly $(2, 0, 1)$ is a 2-algebra; to show that it is h-initial, take any 2-algebra (C, c_0, c_1). By elimination with respect to the constant family C and the elements c_0 and c_1, we have the map $\lambda x.\mathsf{2rec}_{\mathcal{U}_i}(x, c_0, c_1) : 2 \to C$, which is a weak algebra homomorphism by the propositional computation rules. Thus we obtain a term $h : \mathsf{2\text{-}Alg}\big[(2, 0, 1), (C, c_0, c_1)\big]$. Now given any $k : \mathsf{2\text{-}Alg}\big[(2, 0, 1), (C, c_0, c_1)\big]$, we need a term of type $\mathsf{Id}(h, k)$. This term follows from a propositional η-rule, which is derivable by 2-elimination over a suitable identity type.

Conversely, let $(2, 0, 1)$ be an h-initial 2-algebra. To prove elimination, let $x : 2 \vdash C(x) : \mathsf{type}$ with $c_0 : C(0)$ and $c_1 : C(1)$ be given, and consider the 2-algebra (C', c'_0, c'_1) defined by:

$$\begin{aligned} C' &\coloneqq (\Sigma x : 2)C(x)\,, \\ c'_0 &\coloneqq \mathsf{pair}(0, c_0)\,, \\ c'_1 &\coloneqq \mathsf{pair}(1, c_1)\,. \end{aligned}$$

Since 2 is h-initial, there is a map $r : 2 \to C'$ with identities $p_0 : \mathsf{Id}(r0, c'_0)$ and $p_1 : \mathsf{Id}(r1, c'_1)$. Now, we would like to set

$$\mathsf{2rec}_{\mathcal{U}_i}(x, c_0, c_1) = \pi_2(rx) : C(x),$$

where π_2 is the second projection from $C' = (\Sigma x : 2)C(x)$. But recall that in general $\pi_2(z) : C(\pi_1(z))$, and so (taking the case $x = 0$) we have $\pi_2(r0) : C(\pi_1(r0))$ rather than the required $\pi_2(r0) : C(0)$; that is, since it need not be that $\pi_1(r0) = 0$, the term $\pi_2(r0)$ has the wrong type to be $2\mathsf{rec}_{\mathcal{U}_i}(0, c_0, c_1)$. However, we can show that

$$\pi_1 : (\Sigma x : 2)C(x) \to 2$$

is a weak homomorphism, so that the composite $\pi_1 \circ r : (2, 0, 1) \to (2, 0, 1)$ must be propositionally equal to the identity homomorphism $1_2 : (2, 0, 1) \to (2, 0, 1)$, by the contractibility of $2\text{-}\mathsf{Alg}\big[(2, 0, 1), (2, 0, 1)\big]$. Thus there is an identity term $p : \mathsf{Id}(\pi_1 \circ r, 1_2)$, along which we can transport using $p_! : C(\pi_1(r0)) \to C(0)$, thus taking $\pi_2(r0) : C(\pi_1(r0))$ to the term $p_!(\pi_2(r0)) : C(0)$ of the correct type. We can then set

$$2\mathsf{rec}_{\mathcal{U}_i}(x, c_0, c_1) = p_!(\pi_2(rx)) : C(x)$$

to get the required elimination term. The computation rules follow by a rather lengthy calculation. □

Proposition 5.2 is the analogue in Homotopy Type Theory of the characterization of 2 as a strict coproduct $1 + 1$ in extensional type theory. It makes precise the rough idea that, in intensional type theory, 2 is a kind of homotopy coproduct or weak ω-coproduct in the weak ω-category $\mathcal{C}(\mathcal{H})$ of types, terms, identity terms, higher identity terms, ... It is worth emphasizing that h-initiality is a purely type-theoretic notion; despite having an obvious semantic interpretation, it is formulated in terms of inhabitation of specific, definable types. Indeed, Proposition 5.2 and its proof have been completely formalized in the Coq proof assistant [Awodey et al. 2011].

Remark 5.3. A development entirely analogous to the foregoing can be given for the type $\mathbb{N}$ of natural numbers. In somewhat more detail, one introduces the notions of a $\mathbb{N}$-algebra and of a weak homomorphism of $\mathbb{N}$-algebras. Using these, it is possible to define the notion of a homotopy-initial $\mathbb{N}$-algebra, analogue to that of a homotopy-initial 2-algebra in Definition 5.1. With these definitions in place, one can prove an equivalence between the formation, introduction, elimination and propositional computation rules for $\mathbb{N}$ and the existence of a homotopy-initial $\mathbb{N}$-algebra. Here, the propositional computation rules are formulated like those above, *i.e.* by replacing the definitional equalities in the conclusion of the usual computation rules [Nordstrom et al. 2000, Section 5.3] with propositional equalities. We do not pursue this further here, however, since $\mathbb{N}$ can also be presented as a W-type, as we discuss in section 5.6 below.

5.5. The main theorem

Although it is more elaborate to state (and difficult to prove) owing to the presence of recursively generated data, our main result on W-types is analogous to the foregoing example in the following respect: rather than being strict initial algebras, as in the extensional case, weak W-types are instead homotopy-initial algebras. This fact can again be stated entirely syntactically, as an equivalence between two sets of rules: the formation, introduction, elimination, and propositional computation rules (which we spell out below) for W-types, and the existence of an h-initial algebra, in the appropriate sense. Moreover, as in the simple case of the type 2, the proof of the equivalence is again entirely constructive.

The required definitions in the current setting are as follows. Let us assume that

$$x : A \vdash B(x) : \mathsf{type}\,,$$

and define the associated polynomial functor as before:

$$PX = (\Sigma x : A)(B(x) \to X)\,. \tag{4}$$

(Actually, this is now functorial only up to propositional equality, but this change makes no difference in what follows.) By definition, a P-algebra is a type C equipped a function $s_C : PC \to C$. For P-algebras (C, s_C) and (D, s_D), a *weak homomorphism* between them $(f, s_f) : (C, s_C) \to (D, s_D)$ consists of a function $f : C \to D$ and an identity proof

$$s_f : \mathsf{Id}_{PC \to D}\big(f \circ s_C\,, s_D \circ Pf\big)\,,$$

where $Pf : PC \to PD$ is the result of the easily-definable action of P on $f : C \to D$. Such an algebra homomorphism can be represented suggestively in the form:

$$\begin{tikzcd} PC \arrow[r, "Pf"] \arrow[d, "s_C"'] \arrow[dr, phantom, "s_f"] & PD \arrow[d, "s_D"] \\ C \arrow[r, "f"'] & D \end{tikzcd}$$

Accordingly, the type of weak algebra maps is defined by

$$P\text{-}\mathsf{Alg}\big[(C, s_C), (D, s_D)\big] \coloneqq$$
$$(\Sigma f : C \to D)\,\mathsf{Id}(f \circ s_C, s_D \circ Pf)\,.$$

Definition 5.4. A P-algebra (C, s_C) is *homotopy-initial* if for every P-algebra (D, s_D), the type

$$P\text{-}\mathsf{Alg}\big[(C, s_C), (D, s_D)\big]$$

of weak algebra maps is contractible.

Remark 5.5. The notion of h-initiality captures a universal property in which the usual conditions of existence and uniqueness are replaced by conditions of existence and uniqueness up to a system of higher and higher identity proofs. To explain this, let us fix a P-algebra (C, s_C) and assume that it is homotopy-initial. Then, given any P-algebra (D, s_D), there is a weak homomorphism $(f, s_f) : (C, s_C) \to (D, s_D)$, since the type of weak maps from (C, s_C) to (D, s_D), being contractible, is inhabited. Furthermore, for any weak map $(g, s_g) : (C, s_C) \to (D, s_D)$, the contractibility of the type of weak maps implies that there is an identity proof

$$p : \mathsf{Id}\big((f, s_f), (g, s_g)\big)\,,$$

witnessing the uniqueness up to propositional equality of the homomorphism (f, s_f). But it is also possible to prove that the identity proof p is unique up to propositional equality. Indeed, since (f, s_f) and (g, s_g) are elements of a contractible type, the identity type $\mathsf{Id}((f, s_f), (g, s_g))$ is also contractible, as observed in Remark 2.2. Thus, if we have another identity proof $q : \mathsf{Id}((f, s_f), (g, s_g))$, there will be an identity term $\alpha : \mathsf{Id}(p, q)$, which is again essentially unique, and so on. It should also be pointed out that, just as strictly initial algebras are unique up to isomorphism, h-initial algebras are unique up to weak equivalence. It then follows from the Univalence axiom that two h-initial algebras are propositionally equal, a fact that we mention only by the way. Finally, we note that there is also a homotopical version of Lambek's Lemma, asserting that the structure map of an h-initial algebra is itself a weak equivalence, making the algebra a *homotopy fixed point* of the associated polynomial functor. The reader can work out the details from the usual proof and the definition of h-initiality, or consult [Awodey et al. 2011].

The deduction rules that characterize homotopy-initial algebras are obtained from the formation, introduction, elimination and computation rules for W-types stated in Section 5.1 by simply replacing the W-computation rule with the following rule, that we call the propositional W-computation rule.

— Propositional W-computation rule.

$$\frac{\begin{array}{l} w : W \vdash C(w) : \mathsf{type} \\ x : A, u : B(x) \to W, v : (\Pi y : B(x))C(u(y)) \vdash \\ \qquad c(x,u,v) : C(\mathsf{sup}(x,u)) \end{array}}{\begin{array}{l} x : A, u : B(x) \to W \vdash \mathsf{wcomp}(x,u,c) : \\ \qquad \mathsf{Id}\big(\mathsf{wrec}(\mathsf{sup}(x,u),c), c(x,u,\lambda y.\mathsf{wrec}(u(y),c))\big) \end{array}}$$

Remark 5.6. One interesting aspect of this group of rules, to which we shall refer as the *rules for homotopical W-types*, is that, unlike the standard rules for W-types, they are invariant under propositional equality. To explain this more precisely, let us work in a type theory with a type universe $\mathcal{U}$ closed under all the forms of types of $\mathcal{H}$ and W-types. Let $A : \mathcal{U}$, $B : A \to \mathcal{U}$ and define $W := (\mathsf{W}x : A)B(x)$. The invariance of the rules for homotopy W-types under propositional equality can now be expressed by saying that if we have a type $W' : \mathcal{U}$ and an identity proof $p : \mathsf{Id}_U(W, W')$, then the Id-elimination rule implies that W' satisfies the same rules as W, in the sense that there are definable terms playing the role of the primitive constants that appear in the rules for W.

We can now state our main result. Its proof has been formalized in the Coq system, and the proof scripts are available at [Awodey et al. 2011]; thus we provide only an outline of the proof.

THEOREM 5.7. *Over the type theory $\mathcal{H}$, the rules for homotopical W-types are equivalent to the existence of homotopy-initial algebras for polynomial functors.*

PROOF PROOF SKETCH. The two implications are proved separately. First, we show that the rules for homotopical W-types imply the existence of homotopy-initial algebras for polynomial functors. Let us assume that $x : A \vdash B(x) : \mathsf{type}$ and consider the associated polynomial functor P, defined as in (4). Using the W-formation rule, we define $W := (\mathsf{W}x : A)B(x)$ and using the W-introduction rule we define a structure map $s_W : PW \to W$, exactly as in the extensional theory. We claim that the algebra (W, s_W) is h-initial. So, let us consider another algebra (C, s_C) and prove that the type T of weak homomorphisms from (W, s_W) to (C, s_C) is contractible. To do so, observe that the W-elimination rule and the propositional W-computation rule allow us to define a weak homomorphism $(f, s_f) : (W, s_W) \to (C, s_C)$, thus showing that T is inhabited. Finally, it is necessary to show that for every weak homomorphism $(g, s_g) : (W, s_W) \to (C, s_C)$, there is an identity proof

$$p : \mathsf{Id}((f, s_f), (g, s_g))\,. \tag{5}$$

This uses the fact that, in general, a type of the form $\mathsf{Id}((f, s_f), (g, s_g))$, is weakly equivalent to the type of what we call *algebra 2-cells*, whose canonical elements are pairs of the form (e, s_e), where $e : \mathsf{Id}(f, g)$ and s_e is a higher identity proof witnessing the propositional equality between the identity proofs represented by the following pasting diagrams:

$$\begin{tikzcd} PW \arrow[r, bend left, "Pg"] \arrow[d, "s_W"'] \arrow[dr, phantom, "s_g"] & PD \arrow[d, "s_D"] \\ W \arrow[r, bend left, "g"] \arrow[r, bend right, "f"'] \arrow[r, phantom, "e"] & D \end{tikzcd} \qquad \begin{tikzcd} PW \arrow[r, bend left, "Pg"] \arrow[r, bend right, "Pf"'] \arrow[r, phantom, "Pe"] \arrow[d, "s_W"'] \arrow[dr, phantom, "s_f"] & PD \arrow[d, "s_D"] \\ W \arrow[r, bend right, "f"'] & D \end{tikzcd}$$

In light of this fact, to prove that there exists a term as in (5), it is sufficient to show that there is an algebra 2-cell

$$(e, s_e) : (f, s_f) \Rightarrow (g, s_g)\,.$$

The identity proof $e : \mathsf{Id}(f, g)$ is now constructed by function extensionality and W-elimination so as to guarantee the existence of the required identity proof s_e.

For the converse implication, let us assume that the polynomial functor associated to the judgement $x : A \vdash B(x) : \mathsf{type}$ has an h-initial algebra (W, s_W). To derive the W-formation rule, we let $(\mathsf{W}x : A)B(x) := W$. The W-introduction rule is equally simple to derive; namely, for $a : A$ and $t \colon B(a) \to W$, we define $\mathsf{sup}(a, t) : W$ as the result of applying the structure map $s_W \colon PW \to W$ to $\mathsf{pair}(a, t) : PW$. For the W-elimination rule, let us assume its premisses and in particular that $w : W \vdash C(w) : \mathsf{type}$. Using the other premisses, one shows that the type $C := (\Sigma w : W)C(w)$ can be equipped with a structure map $s_C : PC \to C$. By the h-initiality of W, we obtain a weak homomorphism $(f, s_f) : (W, s_W) \to (C, s_C)$. Furthermore, the first projection $\pi_1 : C \to W$ can be equipped with the structure of a weak homomorphism, so that we obtain a diagram of the form

$$\begin{tikzcd} PW \arrow[r, "Pf"] \arrow[d, "s_W"'] & PC \arrow[r, "P\pi_1"] \arrow[d, "s_C"] & PW \arrow[d, "s_W"] \\ W \arrow[r, "f"'] & C \arrow[r, "\pi_1"'] & W\,. \end{tikzcd}$$

But the identity function $1_W : W \to W$ has a canonical structure of a weak algebra homomorphism and so, by the contractibility of the type of weak homorphisms from (W, s_W) to itself, there must be an identity proof between the composite of (f, s_f) with (π_1, s_{π_1}) and $(1_W, s_{1_W})$. This implies, in particular, that there is an identity proof $p : \mathsf{Id}(\pi_1 \circ f, 1_W)$. Since $(\pi_2 \circ f)w : C((\pi_1 \circ f)w)$, we can define

$$\mathsf{wrec}(w, c) := p_!\,((\pi_2 \circ f)w) : C(w)$$

where the transport $p_!$ is defined via Id-elimination over the dependent type

$$u : W \to W \vdash C(u(w)) : \mathsf{type}\,.$$

The verification of the propositional W-computation rule is a rather long calculation, involving several lemmas concerning the naturality properties of operations of the form $p_!$. □

5.6. Definability of inductive types

We conclude this section by indicating how the limited form of extensionality that is assumed in the type theory $\mathcal{H}$, namely the principle of function extensionality, allows us to overcome the obstacles in defining various inductive types as W-types mentioned at the end of Section 5.1, provided that both are understood in the appropriate homotopical way, *i.e.* with all types being formulated with propositional computation rules.

Consider first the paradigmatic case of the type of natural numbers. To define it as a W-type, we work in an extension of the type theory $\mathcal{H}$ with

— formation, introduction, elimination and propositional computation rules for types $\mathsf{0}$, $\mathsf{1}$ and $\mathsf{2}$ that have zero, one and two canonical elements, respectively;
— the rules for homotopy W-types, as stated above;
— rules for a type universe $\mathcal{U}$ reflecting all the forms of types of $\mathcal{H}$, W-types, and $\mathsf{0}$, $\mathsf{1}$ and $\mathsf{2}$.

In particular, the rules for 2 are those given in Section 5.4. We then proceed as follows. We begin by setting $A = 2$, as in the extensional case. We then define a dependent type

$$x : A \vdash B(x) : \mathcal{U}$$

by 2-elimination, so that the propositional 2-computation rules give us propositional equalities

$$p_0 : \mathsf{Id}_U(0, B(0))\,, \qquad p_1 : \mathsf{Id}_U(1, B(1))\,.$$

Because of the invariance of the rules for 0 and 1 under propositional equalities (as observed in Remark 5.6), we can then derive that the types $B(0)$ and $B(1)$ satisfy rules analogous to those for 0 and 1, respectively. This allows us to show that the type

$$\mathbb{N} := (\mathsf{W}x : A)B(x)$$

satisfies the introduction, elimination and propositional computation rules for the type of natural numbers. The proof of this fact proceeds essentially as one would expect, but to derive the propositional computation rules it is useful to observe that for every type $X : \mathcal{U}$, there are adjoint homotopy equivalences, in the sense of Definition 2.3, between the types $0 \to X$ and 1, and between $1 \to X$ and X. Indeed, the propositional identities witnessing the triangular laws are useful in the verification of the propositional computation rules for $\mathbb{N}$. For details, see the formal development in Coq provided in [Awodey et al. 2011]. Observe that as a W-type, $\mathbb{N}$ is therefore also an h-initial algebra for the equivalent polynomial functor $P(X) = 1 + X$, as expected.

Finally, let us observe that the definition of a type representing the second number class as a W-type, as discussed in [Martin-Löf 1984], carries over equally well. Indeed, one now must represent type-theoretically a signature with three operations: the first of arity zero, the second of arity one, and the third of arity $\mathbb{N}$. For the first two we can proceed exactly as before, while for the third there is no need to prove auxiliary results on adjoint homotopy equivalences. As before, the second number class supports an h-initial algebra structure for the corresponding polynomial functor $P(X) = 1 + X + (\mathbb{N} \to X)$. Again, the formal development of this result in Coq can be found in [Awodey et al. 2011].

6. FUTURE WORK

The treatment of W-types presented here is part of a larger investigation of general inductive types in Homotopy Type Theory. We sketch the projected course of our further research.

(1) In the setting of extensional type theory, Dybjer [Dybjer 1997] showed that every strictly positive definable functor can be represented as a polynomial functor, so that all such inductive types are in fact W-types. This result should generalize to the present setting in a straightforward way.

(2) Also in the extensional setting, Gambino and Hyland [Gambino and Hyland 2004] showed that general tree types [Petersson and Synek 1989] [Nordstrom et al. 1990, Chapter 16], viewed as initial algebras for general polynomial functors, can be constructed from W-types in locally cartesian closed categories, using equalizers. We expect this result to carry over to the present setting as well, using Id-types in place of equalizers.

(3) In [Voevodsky 2009] Voevodsky has shown that all inductive types of the Predicative Calculus of Inductive Constructions can be reduced to the following special cases:
   - 0, 1, $A + B$, $(\Sigma x : A)B(x)$,
   - $\mathsf{Id}_A(a, b)$,

— general tree types.

Combining this with the foregoing, we expect to be able to extend our Theorem 5.7 to the full system of predicative inductive types underlying Coq.

Finally, one of the most exciting recent developments in Univalent Foundations is the idea of Higher Inductive Types (HITs), which can also involve identity terms in their signature [Lumsdaine 2011; Shulman 2011]. This allows for algebras with equations between terms, like associative laws, coherence laws, etc.; but the really exciting aspect of HITs comes from the homotopical interpretation of identity terms as paths. Viewed thus, HITs should permit direct formalization of many basic geometric spaces and constructions, such as the unit interval I; the spheres S^n, tori, and cell complexes; truncations, such as the [bracket] types [Awodey and Bauer 2004]; various kinds of quotient types; homotopy (co)limits; and many more fundamental and fascinating objects of geometry not previously captured by type-theoretic formalizations. Our investigation of conventional inductive types in the homotopical setting should lead to a deeper understanding of these new and important geometric analogues.

ACKNOWLEDGMENTS

We would like to thank Vladimir Voevodsky and Michael Warren for helpful discussions on the subject of this paper. In particular, Vladimir Voevodsky suggested a simplification of the proof that the rules for homotopical W-types imply h-initiality.

Steve Awodey gratefully acknowledges the support of the National Science Foundation, Grant DMS-1001191 and the Air Force OSR, Grant 11NL035. Nicola Gambino is grateful for the support of the Institute for Advanced Study, where he worked on this project. This work was supported by the National Science Foundation under agreement No. DMS-0635607. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Kristina Sojakova is grateful for the support of CyLab at Carnegie Mellon under grants DAAD19-02-1-0389 and W911NF-09-1-0273 from the Army Research Office.

REFERENCES

M. Abbott, T. Altenkirch, and N. Ghani. 2005. Containers: Constructing strictly positive types. *Theoretical Computer Science* 342, 1 (2005), 3–27.

T. Altenkirch, C. McBride, and W. Swierstra. 2007. Observational equality, now. In *Programming languages meets program verification (PLPV '07)*. ACM, 57–68.

S. Awodey. 2010. Type theory and homotopy. (2010). Available from the author's web page.

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

S. Awodey, N. Gambino, and K. Sojakova. 2011. Inductive types in Homotopy Type Theory: Coq proofs. (2011). Available at: www.andrew.cmu.edu/$\sim$awodey/hott/ithottCoq.zip.

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

M. Batanin. 1998. Monoidal globular categories as a natural environment for the theory of weak n-categories. *Advances in Mathematics* 136, 1 (1998), 39–103.

Y. Bertot and P. Castéran. 2004. *Interactive Theorem Proving and Program Development. Coq'Art: the Calculus of Inductive Constructions*. Springer-Verlag.

T. Coquand and C. Paulin-Mohring. 1990. Inductively defined types. In *International Conference on Computer Logic (COLOG '88) (LNCS)*, Vol. 417. Springer.

P. Dybjer. 1997. Representing inductively defined sets by well-orderings in Martin-Löf's type theory. *Theoretical Computer Science* 176 (1997), 329–335.

N. Gambino and M. Hyland. 2004. Well-founded trees and dependent polynomial functors. In *Types for Proofs and Programs (TYPES '03) (LNCS)*, S. Berardi, M. Coppo, and F. Damiani (Eds.), Vol. 3085. 210–225.

R. Garner. 2009. On the strength of dependent products in the type theory of Martin-Löf. *Annals of Pure and Applied Logic* 160 (2009), 1–12.

H. Goguen and Z. Luo. 1993. Inductive data types: well-ordering types revisited. In *Logical Environments*, G. Huet and G. Plotkin (Eds.). Cambridge University Press, 198–218.

E. Griffor and M. Rathjen. 1994. The strength of some Martin-Löf type theories. *Archive for Mathematical Logic* 33, 5 (1994), 347–385.

M. Hofmann. 1997. *Extensional constructs in intensional type theory*. Springer-Verlag.

M. Hofmann and T. Streicher. 1998. The groupoid interpretation of type theory. In *Twenty-five years of constructive type theory 1995 (Oxford Logic Guides)*, Vol. 36. Oxford Univ. Press, 83–111.

W. Howard. 1980. The formulae-as-types notion of construction. In *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, J. Seldin and J. Hindley (Eds.). Academic Press, 479–490.

T. Leinster. 2004. *Higher operads, higher categories*. Cambridge University Press.

P. Lumsdaine. 2010a. *Higher categories from type theories*. Ph.D. Dissertation. Carnegie Mellon University.

P. Lumsdaine. 2010b. Weak ω-categories from intensional type theory. *Logical Methods in Computer Science* 6 (2010), 1–19.

P. Lumsdaine. 2011. Higher inductive types: a tour of the menagerie. (2011). Post on the Homotopy Type Theory blog.

P. Martin-Löf. 1975. An Intuitionistic Theory of Types: Predicative Part. In *Logic Colloquium 1973*, H. Rose and J. Shepherdson (Eds.). North-Holland, 73–118.

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

P. Martin-Löf. 1984. Intuitionistic Type Theory. Notes by G. Sambin of a series of lectures given in Padua, 1980. (1984). Bibliopolis.

C. McBride. 2010. W-types: good news and bad news. (2010). Post on the Epigram blog.

C. McBride and J. McKinna. 2004. The view from the left. *Journal of Functional Programming* 14, 1 (2004), 16–111.

I. Moerdijk and E. Palmgren. 2000. Well-founded trees in categories. *Annals of Pure and Applied Logic* 104 (2000), 189–218.

B. Nordstrom, K. Petersson, and J. Smith. 1990. *Programming in Martin-Löf type theory*. Oxford University Press.

B. Nordstrom, K. Petersson, and J. Smith. 2000. Martin-Löf type theory. In *Handbook of Logic in Computer Science*, Vol. 5. Oxford University Press, 1–37.

U. Norell. 2007. *Towards a Practical Programming Language Based on Dependent Type Theory*. Ph.D. Dissertation. Chalmers University of Technology.

C. Paulin-Mohring. 1993. Inductive definitions in the system Coq - rules and properties. In *Typed Lambda Calculi and Applications (TLCA '93)*, Vol. 664. Springer.

K. Petersson and D. Synek. 1989. A set constructor for inductive sets in Martin-Löf type theory. In *Category Theory and Computer Science (LNCS)*, Vol. 389. Springer-Verlag.

M. Shulman. 2011. Homotopy Type Theory, VI. (2011). Post on the n-category cafe blog.

T. Streicher. 1993. Investigations into intensional type theory. (1993). Habilitation Thesis. Available from the author's web page.

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

B. van den Berg and R. Garner. 2012. Topological and Simplicial Models of Identity Types. *ACM Transactions on Computational Logic* 13, 1 (2012).

V. Voevodsky. 2009. Notes on type systems. (2009). Available from the author's web page.

V. Voevodsky. 2010a. Univalent foundations Coq files. (2010). Available from the author's web page.

V. Voevodsky. 2010b. Univalent foundations project. (2010). Available from the author's web page.

Received ; revised ; accepted