Article:
Awodey, S, Gambino, N orcid.org/0000-0002-4257-3590 and Sojakova, K (2017) Homotopy-initial algebras in type theory. Journal of the ACM, 63 (6). ISSN 1535-9921

https://doi.org/10.1145/3006383

Homotopy-initial algebras in type theory

Journal:	*Journal of the ACM*
Manuscript ID	JACM-00072-2015.R1
Manuscript Type:	Paper
Date Submitted by the Author:	04-Oct-2015
Complete List of Authors:	Awodey, Steve Gambino, Nicola Sojakova, Kristina
Computing Classification Systems:	Theory of computation, Logic, Type Theory

HOMOTOPY-INITIAL ALGEBRAS IN TYPE THEORY

STEVE AWODEY, NICOLA GAMBINO, AND KRISTINA SOJAKOVA

Abstract. We investigate inductive types in type theory, using the insights provided by homotopy type theory and univalent foundations of mathematics. We do so by introducing the new notion of a homotopy-initial algebra. This notion is defined by a purely type-theoretic contractibility condition which replaces the standard, category-theoretic universal property involving the existence and uniqueness of appropriate morphisms. Our main result characterises the types that are equivalent to W-types as homotopy-initial algebras.

Introduction

Inductive types, such as the type of natural numbers and types of well-founded trees, are one of the fundamental ingredients of dependent type theories, including Martin-Löf's type theories [29] and the Calculus of Inductive Constructions [7, 10]. In the present work, we investigate inductive types using the insights provided by homotopy type theory [31] and univalent foundations of mathematics [33].

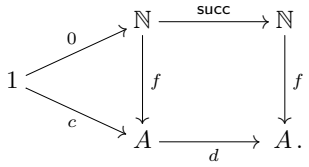
As an introduction to the general problem that we will investigate, let us consider the case of the type of natural numbers. Its elimination rule can be seen as the propositions-as-types translation of the familiar induction principle:

$$\frac{x:\mathbb{N}\vdash E(x):\mathsf{type}\quad c:E(0)\quad x:\mathbb{N},y:E(x)\vdash d(x,y):E(\mathsf{succ}(x))}{x:\mathbb{N}\vdash \mathsf{elim}(x,c,d):E(x)\,.} \tag{E}$$

Here, E is considered as a predicate on the type $\mathbb{N}$, c as a proof that E holds for 0, and d as a program that transforms a proof y that E holds for $x:\mathbb{N}$ into a proof $d(x,y)$ that E holds for the successor $\mathsf{succ}(x)$. As is well-known, the special case of the rule (E) obtained by considering the dependent type in its premiss to be constant provides a counterpart of the familiar principle of defintion of a function by recursion:

$$\frac{A:\mathsf{type}\quad c:A\quad y:A\vdash d(y):A}{x:\mathbb{N}\vdash \mathsf{rec}(x,c,d):A\,.} \tag{R}$$

This rule is closely related to Lawvere's notion of a natural number object in a category [21]. Indeed, it allows us to define a function $f:\mathbb{N}\to A$ such that each face in the following diagram commutes:



Within type theory, the commutativity of the diagram is expressed by judgemental equalities

$$f(0)=c:A\,,\quad x:\mathbb{N}\vdash f(\mathsf{succ}(x))=d(f(x)):A\,,$$

Date: October 4, 2015.

which can be proved as a special case of the computation rules for $\mathbb{N}$.

In the notion of a natural number object, however, one not only requires the existence of such a function f, but also its uniqueness. Remarkably, within type theories, it is possible to use the elimination rule (E) to show that such a function f is unique up to a pointwise propositional equality, i.e. that given another function $g : \mathbb{N} \to A$ making the corresponding diagram commute, there are propositional equalities $\phi_x : \mathsf{Id}_A(fx, gx)$ for every $x : \mathbb{N}$. This suggests the possibility of characterizing inductive types, such as the type of natural numbers, by means of standard category-theoretic universal properties. Unfortunately, this seems to be possible only in the presence of additional extensionality principles such as the equality reflection rule (which forces propositional equality to coincide with judgemental equality) [11, 14, 27]. Without these principles, the uniqueness up to pointwise propositional equality of the functions defined by recursion does not seem to be sufficient to derive the elimination and computation rules for inductive types. Indeed, the elimination rules imply not only the existence of pointwise propositional equalities, as above, but also their essential uniqueness, expressed by a system of higher and higher propositional equalities whose combinatorics are difficult to axiomatize directly.

The aim of this paper is to solve this problem using ideas inspired by the recent connections between type theory, homotopy theory and higher-dimensional category theory [4, 5, 12, 19, 23], which are at the core of homotopy type theory [31] and Voevodsky's univalent foundations of mathematics programme [32]. Our analysis focuses on well-ordering types (W-types for short), which can be easily characterized as initial algebras for polynomial functors within extensional type theories [1, 11, 13, 27]. Our results show that in the system under consideration, a type is equivalent to a W-type if and only if it is a *homotopy-initial algebra* for a polynomial functor. The notion of homotopy-initial algebra, which we introduce here, is intended as a generalization of the standard category-theoretic notion of an initial algebra, obtained by replacing the usual existence and uniqueness requirements by asking for the *contractibility* of suitable types of algebra morphisms. The notion of homotopy-initial algebra is entirely type-theoretic, but it is inspired by ideas of higher-dimensional category theory, where standard category-theoretic universal properties are generalized using the topological notion of contractibility [24]. Our account of homotopy-initial algebras is entirely syntactic, and we expect future semantic accounts would be given using homotopy-invariant versions of initial algebras for polynomial functors (cf. [6]).

As part of our development, we also establish several results that do not have counterparts in the extensional setting. For example, we show how the elements of the identity type between two algebra morphisms are essentially type-theoretic counterparts of the notion of an algebra 2-cell [8]. This surprising fact provides further evidence for the idea that the rules for identity types encapsulate some higher-dimensional categorical structure [5, 23]. We also analyze the complexity of the types of proofs that a given type is homotopy-initial, showing that it is a *mere proposition*, i.e. a type of homotopy level 1 [33]. Finally, we show that, under the assumption of Voevodsky's univalence axiom, a version of univalence also holds for algebras and that such algebras, when they exist, are essentially unique, i.e. unique up to a contractible type of propositional equalities. It may be noted that, because of the higher-dimensional structure provided by identity types, polynomial functors may acquire further aspects not present in the extensional setting, since the dependent types that determine polynomial functors may have homotopy level greater than 2 (cf. [20]).

Our development can be extended without difficulty to other kinds of inductive types, such as coproducts $A + B$ and the natural numbers $\mathbb{N}$. In fact, in order to illustrate our ideas, we begin the paper by considering the simpler case of the type Bool of Boolean truth values, establishing analogues of the results proved later for W-types.

Some of the results presented here were announced in our extended abstract [3], and are summarized in the book [31]. The present paper expands the material outlined there by including

not only all of the omitted proofs (which requires the statement of auxiliary lemmas), but also a new, more algebraic treatment of the elimination and computation rules for inductive types, as well as an analysis of the complexity of the type of proofs that a type is homotopy-initial, and an investigation of the further consequences of the univalence axiom.

Formalization. All the results in this paper have been formally proved in the current version of the Coq proof assistant. The formalization files, which build on the existing libraries for homotopy type theory and univalent foundations of mathematics, are available from the third author's GitHub repository:

`https://github.com/kristinas/hinitiality`

One should note that the type theory underlying the current version of Coq differs from that used in this paper in several respects. Although the formalization uses only a small fragment of the type theory implemented in Coq, it does use its judgemental η-rules for both Π-types and Σ-types (see below for details). The results in this paper are obtained working instead in a type theory that assumes the judgemental η-rules for Π-types, but not for Σ-types.

Organization of the paper. Section 1 reviews all of the preliminaries necessary to read the paper and introduces the type theory $\mathcal{H}$ which will provide the background theory for our investigations. The rest of the paper is divided in two parts. The first part considers the type Bool. We begin in Section 2 by defining the notions of a bipointed type, bipointed morphism, fibered bipointed type, bipointed section analyzing homotopies between morphisms and sections in terms of identity types. We also discuss the notion of equivalence between bipointed types. Section 3 introduces the notions of inductive bipointed type and homotopy-initial bipointed type, so as to arrive at the main results, characterizing Bool up to equivalence and exploring consequences of the univalence axiom. The second part, which comprises Sections 4 and 5, proceeds in parallel with the first part, but with algebras for a polynomial functor instead of bipointed types. This second part forms the main contribution of the paper, while the first part provides a simpler setting in which to introduce the new concepts and methods of proof. The formal structure of the two parts is intentionally parallel, in order to guide the reader through the more difficult, second part.

1. Homotopy-theoretic concepts in type theory

Review of type theory. The type theories considered in this paper are formulated using the following four 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 *judgemental equality*, which should be contrasted with the notion of *propositional equality* defined below. Each kind of judgement can also be made relative to a context of variable declarations Γ, e.g. $\Gamma \vdash A : \mathsf{type}$. However, when stating deduction rules we may omit the mention of a context common to premisses and conclusions of the rule, and we make use of other standard conventions to simplify the exposition.

We begin by introducing a very basic version of Martin-Löf's type theory, denoted by $\mathcal{M}$. This type theory has rules for the following forms of type:

$$(\Sigma x : A)B(x)\,, \quad (\Pi x : A)B(x)\,, \quad \mathsf{Id}_A(a,b)\,, \quad \mathsf{U}\,.$$

The rules for these types are recalled in Tables 1, 2, 3 and 4, respectively. The rules are as in [29], except that the rules for the type universe U are stated à la Russell for simplicity. As usual, we refer to an element of the form appearing in the conclusion of an introduction rule as a *canonical element*.

Let us establish some notation and recall some basic facts and terminogy. First of all, for $f : (\Pi x : A)B(x)$ and $a : A$, we write $f(a)$ or fa instead of $\mathsf{app}(f, a)$. We may also write (a, b) instead of $\mathsf{pair}(a, b)$ to denote canonical elements of Σ-types. Given types A and B, the product

$$\frac{x : A \vdash B(x) : \mathsf{type}}{(\Sigma x : A)B(x) : \mathsf{type}} \qquad \frac{a : A \qquad b(a) : B(a)}{\mathsf{pair}(a, b) : (\Sigma x : A)B(x)}$$

$$\frac{z : (\Sigma x : A)B(x) \vdash E(z) : \mathsf{type} \quad x : A, y : B(x) \vdash e(x, y) : E(\mathsf{pair}(x, y))}{z : (\Sigma x : A)B(x) \vdash \mathsf{split}(z, e) : E(z)}$$

$$\frac{z : (\Sigma x : A)B(x) \vdash E(z) : \mathsf{type} \quad x : A, y : B(x) \vdash e(x, y) : E(\mathsf{pair}(x, y))}{x : A, y : B(x) \vdash \mathsf{split}(\mathsf{pair}(x, y), e) = e(x, y) : E(\mathsf{pair}(x, y))}$$

TABLE 1. Rules for Σ-types.

$$\frac{x : A \vdash B(x) : \mathsf{type}}{(\Pi x : A)B(x) : \mathsf{type}} \qquad \frac{x : A \vdash b(x) : B(x)}{(\lambda x : A)b(x) : (\Pi x : A)B(x)}$$

$$\frac{f : (\Pi x : A)B(x) \quad a : A}{\mathsf{app}(f, a) : B(a)} \qquad \frac{x : A \vdash b(x) : B(x)}{\mathsf{app}((\lambda x : A)b(x), a) = b(a) : B(a)}$$

TABLE 2. Rules for Π-types.

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

$$\frac{x, y : A, u : \mathsf{Id}_A(x, y) \vdash E(x, y, u) : \mathsf{type} \qquad x : A \vdash e(x) : E(x, x, \mathsf{refl}(x))}{x, y : A, u : \mathsf{Id}_A(x, y) \vdash \mathsf{J}(x, y, u, e) : E(x, y, u)}$$

$$\frac{x, y : A, u : \mathsf{Id}_A(x, y) \vdash E(x, y, u) : \mathsf{type} \qquad x : A \vdash e(x) : E(x, x, \mathsf{refl}(x))}{x : A \vdash \mathsf{J}(x, x, \mathsf{refl}(x), e) = e(x) : E(x, x, \mathsf{refl}(x))}$$

TABLE 3. Rules for Id-types.

$$\frac{A : \mathsf{U} \quad x : A \vdash B(x) : \mathsf{U}}{(\Sigma x : A)B(x) : \mathsf{U}} \qquad \frac{A : \mathsf{U} \quad x : A \vdash B(x) : \mathsf{U}}{(\Pi x : A)B(x) : \mathsf{U}}$$

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

TABLE 4. Rules for the type universe U.

type $A \times B$ and the function $A \to B$ are defined via Σ-types and Π-types in the usual way. As is standard, we let $A \leftrightarrow B =_{\text{def}} (A \to B) \times (B \to A)$. The rules for Σ-types allow us to derive the rules for projections

$$\frac{c : (\Sigma x : A)B(x)}{\pi_1(c) : A} \qquad \frac{c : (\Sigma x : A)B(x)}{\pi_2(c) : B(\pi_1(c))}\,.$$

We say that two elements $a, b : A$ are *propositionally equal* if the type $\mathsf{Id}_A(a, b)$ is inhabited and write $a \cong b$ to denote this situation. The rules for Σ-types allow us to prove the following propositional form of the η-rule for Σ-types:

$$\frac{c : (\Sigma x : A)B(x)}{\eta_c : \mathsf{Id}(c, \mathsf{pair}(\pi_1(c), \pi_2(c)))\,.} \tag{1.1}$$

This rule asserts that every element of a Σ-type is propositionally equal to one of canonical form. Note that none of the type theories we consider in this paper include the judgemental form of the η-rules for Σ-types, as is done in [14]. The presence of the type universe U allows us to define the notion of a small type: as usual, we say that a type A is *small* if it is an element of the type universe, i.e. $A : \mathsf{U}$.

We write $\mathcal{M}^{\mathrm{ext}}$ for the extensional type theory obtained from $\mathcal{M}$ by adding the following rule, known as the *identity reflection rule*:

$$\frac{p : \mathsf{Id}_A(a, b)}{a = b : A\,.} \tag{1.2}$$

This rule collapses propositional equality to definitional equality, thus making the overall system somewhat simpler to work with, but makes type-checking undecidable [15]. For this reason, it is not assumed in the most recent formulations of Martin-Löf type theories [29] or in automated proof assistants like Coq [7]. Rather than working in $\mathcal{M}^{\mathrm{ext}}$, we work in a weaker extension of $\mathcal{M}$ which we now describe.

The type theory $\mathcal{H}$. The type theory $\mathcal{H}$ which will serve as the background theory for our development extends the type theory $\mathcal{M}$ described above with two additional rules. The first additional rule is a judgemental form of the η-rule for Π-types:

$$\frac{f : (\Pi x : A)B(x)}{f = (\lambda x : A)\mathsf{app}(f, x) : (\Pi x : A)B(x)\,.} \tag{1.3}$$

An immediate consequence of this rule is that we can identify a family of small types, given by a dependent type $x : A \vdash B(x) : \mathsf{U}$ with functions $B : A \to \mathsf{U}$. In the following, we shall refer to both of these as *small dependent types*. The second additional rule is the *function extensionality* axiom, which is considered here with propositional equalities:

$$\frac{f : (\Pi x : A)B(x) \qquad g : (\Pi x : A)B(x) \qquad x : A \vdash \alpha_x : \mathsf{Id}_{B(x)}(f(x), g(x))}{\mathsf{funext}(f, g, \alpha) : \mathsf{Id}_{(\Pi x\,:\,A)B(x)}(f, g)\,.} \tag{1.4}$$

Note that $\mathcal{H}$ does not have any ground types apart from the type universe U. This is because these type theories are intended as background theories for our study of inductive types. The type theory $\mathcal{H}$ does not include any global extensionality principles, like the identity reflection rule, the K rule, or the uniqueness of identity proofs (UIP) principle [30]. This makes it possible for $\mathcal{H}$ to have not only straightforward set-theoretic models (where those extensionality principles are valid), but also with homotopy-theoretic models, such as the groupoid model [16] and the simplicial model [19], in which the rules of $\mathcal{H}$, but not the extensionality principles mentioned above, remain valid. Indeed, $\mathcal{H}$ is a subsystem of the type theory used in Voevodsky's univalent foundations of Mathematics programme [33]. In particular, the function extensionality axiom in (1.4) is formally implied by the univalence axiom [32] (using the fact that function extensionality, as stated in (1.4), follows from its special case for function types). But, in contrast with the

univalence axiom, the function extensionality axiom is valid also in set-theoretic models. Uses of the univalence axiom will be explicitely noted.

We write $\mathcal{H}^{\mathrm{ext}}$ for the extension of $\mathcal{H}$ with the identity reflection rule in (1.2).

Remark. Our results continue to hold when the judgemental η-rule for Π-types in (1.3) is weakened by replacing the judgemental equality in its conclusion with a propositional one, which is derivable if Π-types are defined as inductive types, as done in [28]. However, since some of our proofs can be simplified in its presence and the current version of the Coq proof assistant assumes the rule (1.3), we prefer to work with it in order to keep our presentation simpler and closer to the formalization.

Homotopy-theoretic notions in type theory. For the convenience of the reader, we review some ideas developed in more detail in [31, 32]. First of all, we will frequently refer to elements of identity types of the form $p : \mathsf{Id}_A(a, b)$ as *paths* (from a to b in A). By the Id-elimination rules, for every dependent type

$$x : A \vdash E(x) : \mathsf{type}\,, \tag{1.5}$$

a path $p : \mathsf{Id}_A(a, b)$ determines the so-called *transport functions*

$$p_! : E(a) \to E(b)\,, \quad p^* : E(b) \to E(a)\,.$$

These are defined so that, for $x : A$, the functions $\mathsf{refl}(x)_!$ and $\mathsf{refl}(x)^*$ are definitionally equal to the identity function $1_{E(x)} : E(x) \to E(x)$. In order to emphasize the fact that dependent types are interpreted as fibrations in homotopy-theoretic models, we sometimes refer to a dependent type as in (1.5) as a *fibered type* over A. Accordingly, elements of the type $(\Pi x : A)E(x)$ may be referred to as *sections* of the fibered type. This terminology is supported by the fact that a section $f : (\Pi x : A)E(x)$ determines a function $f' : A \to E'$, where $E' =_{\mathrm{def}} (\Sigma x : A)E(x)$, defined by $f' =_{\mathrm{def}} (\lambda x : A)\mathsf{pair}(x, fx)$, which is such that $\pi_1 f'(x) = x$ for every $x : A$. We represent such a situation with the diagram

$$\begin{array}{ccc} E' & & \\ {\scriptstyle \pi_1}\downarrow & \nwarrow & \\ A\,. & & {\scriptstyle f'} \end{array}$$

Let us now review the notion of an equivalence of types. In order to do this, we need some auxiliary notions. Recall that a type A is said to be *contractible* if the type

$$\mathsf{iscontr}(A) =_{\mathrm{def}} (\Sigma x : A)(\Pi y : A)\mathsf{Id}_A(x, y) \tag{1.6}$$

is inhabited. The type $\mathsf{iscontr}(A)$ 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. Next, we define the *homotopy fiber* of a function $f : A \to B$ over $y : B$ as the type

$$\mathsf{hfiber}(f, y) =_{\mathrm{def}} (\Sigma x : A)\,\mathsf{Id}_B(fx, y)\,.$$

A function $f : A \to B$ is then said to be an *equivalence* if and only if all of its homotopy fibers are contractible, i.e. the type

$$\mathsf{isequiv}(f) =_{\mathrm{def}} (\Pi y : B)\,\mathsf{iscontr}(\mathsf{hfiber}(f, y))$$

is inhabited. The notion of an equivalence was defined in [33] and is inspired by homotopy-theoretic ideas. It is of particular importance since it provides a fully internal notion of isomorphism between types. For types A and B, the type $\mathsf{Equiv}(A, B)$ of equivalences from A to B is

defined so that its canonical elements are pairs consisting of a function $f : A \to B$ and a proof that it is an equivalence, i.e. we let

$$\mathsf{Equiv}(A, B) =_{\mathrm{def}} (\Sigma f : A \to B)\, \mathsf{isequiv}(f)\,. \tag{1.7}$$

We write $A \simeq B$ if there is an equivalence from A to B. For example, the well-known $\Pi\Sigma$-distributivity, which is sometimes referred to as the *type-theoretic axiom of choice* [26], can be expressed as an equivalence

$$(\Pi x : A)(\Sigma y : B(x))E(x, y) \simeq (\Sigma u : (\Pi x : A)B(x))(\Pi x : A)E(x, ux)\,. \tag{1.8}$$

It can be shown within the type theory $\mathcal{H}$ that a function $f : A \to B$ is an equivalence if and only if it has a two-sided inverse, i.e. there exists a function $g : B \to A$ such that the types $\mathsf{Id}(gf, 1_A)$ and $\mathsf{Id}(fg, 1_B)$ are inhabited. However, the type of equivalences is not equivalent to the type of functions with a two-sided inverse as above, but instead (as suggested by André Joyal, cf. [18, Definition 3.1.1]) to the type of functions that have a left inverse and a right inverse, i.e. functions $g : B \to A$ and $h : B \to A$ such that the types $\mathsf{Id}(gf, 1_A)$ and $\mathsf{Id}(fh, 1_B)$ are inhabited. More precisely, for every $f : A \to B$, there is an equivalence

$$\mathsf{isequiv}(f) \simeq \big((\Sigma g : B \to A)\mathsf{Id}(gf, 1_A) \times (\Sigma h : B \to A)\mathsf{Id}(fh, 1_B)\big)\,. \tag{1.9}$$

For our purposes, the idea of equivalences as functions with a left and a right inverse will be most easily generalized when we consider types equipped with additional structure.

Because of the presence of the principle of function extensionality in $\mathcal{H}$, identity types of function types and of Π-types admit an equivalent description in terms of the notion of a homotopy, which we now review. For $f\,, g : (\Pi x : A)B(x)$, the type of homotopies between f and g is defined by letting

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

We sometimes write $\alpha : f \sim g$ rather than $\alpha : \mathsf{Hot}(f, g)$.

One of the key insights derived from the homotopy-theoretic interpretation of type theories is that the notion of contractibility in (1.6) can be used to articulate the world of types into a hierarchy of so-called *homotopy levels* (or h-levels for short) according to their homotopical complexity [32]. These are defined inductively by saying that a type A has level 0 if it is contractible and it has level $n + 1$ if for every $x, y : A$ the type $\mathsf{Id}_A(x, y)$ has level n. Types of h-level 1 are called here *mere propositions*. By definition, a type A is said to be a *mere proposition* if the type

$$\mathsf{isprop}(A) =_{\mathrm{def}} (\Pi x : A)(\Pi y : A)\, \mathsf{iscontr}(\mathsf{Id}_A(x, y))$$

is inhabited.

Characterization of identity types. We now recall that the identity types of various kinds of compound types admit an equivalent description. We begin by considering product types and function types. Let A and B be types. For any $c, d : A \times B$, and any $f, g : A \to B$, we have canonical maps

$$\begin{aligned}
&\mathsf{ext}^{\times}_{c,d} : \mathsf{Id}_{A\times B}(c, d) \to \mathsf{Id}_A(\pi_1 c, \pi_1 d) \times \mathsf{Id}_B(\pi_2 c, \pi_2 d)\,,\\
&\mathsf{ext}^{\to}_{f,g} : \mathsf{Id}_{A\to B}(f, g) \to (\Pi x : A)\mathsf{Id}_B(fx, gx)\,.
\end{aligned}$$

Note that the codomain of the second map is $\mathsf{Hot}(f, g)$. These functions can be easily generalized to Σ-types and Π-types, so as to obtain functions

$$\begin{aligned}
&\mathsf{ext}^{\Sigma}_{c,d} : \mathsf{Id}_{(\Sigma x : A)B(x)}(c, d) \to (\Sigma p : \mathsf{Id}_A(\pi_1 c, \pi_1 d))\, \mathsf{Id}_{B(\pi_2 d)}(p_!(\pi_2 c), \pi_2 d)\,,\\
&\mathsf{ext}^{\Pi}_{f,g} : \mathsf{Id}_{(\Pi x : A)B(x)}(f, g) \to (\Pi x : A)\mathsf{Id}_{B(x)}(fx, gx)\,.
\end{aligned}$$

Again, the codomain of the second map is $\mathsf{Hot}(f,g)$. Furthermore, for the type universe U, there is an evident function

$$\mathsf{ext}^{\mathsf{U}}_{A,B} : \mathsf{Id}_{\mathsf{U}}(A,B) \to \mathsf{Equiv}(A,B)\,.$$

We refer to these functions as the *extension functions* for product types, function types, Σ-types, Π-types and U, respectively. We then have that the extension functions for product types and Σ-types can be shown to be equivalences within the type theory $\mathcal{M}$, using the (provable) η-rule for Σ-types in (1.1). The extension functions for function types and Π-types, for their part, can be shown to be equivalences within the type theory $\mathcal{H}$, using the function extensionality principle in (1.4) that is part of $\mathcal{H}$. Finally, the assertion that the extension function for the type universe is an equivalence is exactly the univalence axiom. Thus, within the type theory $\mathcal{H}$ we have the following inverses to the extension functions

$$\begin{aligned}
&\mathsf{int}^{\times}_{c,d} : \big(\mathsf{Id}_A(\pi_1 c\,, \pi_1 d) \times \mathsf{Id}_B(\pi_2 c, \pi_2 d)\big) \to \mathsf{Id}_{A\times B}(c,d)\\
&\mathsf{int}^{\to}_{f,g} : \big((\Pi x : A)\mathsf{Id}_B(fx, gx)\big)\big) \to \mathsf{Id}_{A\to B}(f,g)\\
&\mathsf{int}^{\Sigma}_{c,d} : \big((\Sigma p : \mathsf{Id}_A(\pi_1 c, \pi_1 d))\mathsf{Id}_{B(\pi_2 c)}(p_!\pi_2 c, \pi_2 d)\big) \to \mathsf{Id}_{(\Sigma x\,:\,A)B(x)}(c,d)\\
&\mathsf{int}^{\Pi}_{f,g} : (\Pi x : A)\mathsf{Id}_{B(x)}(fx, gx) \to \mathsf{Id}_{(\Pi x\,:\,A)B(x)}(f,g)\,,
\end{aligned}$$

and, in the extension of $\mathcal{H}$ with the univalence axiom, also the inverse

$$\mathsf{int}^{\mathsf{U}}_{A,B} : \mathsf{Id}_{\mathsf{U}}(A,B) \to \mathsf{Equiv}(A,B)\,.$$

In the following, if the context does not create any confusion, we may omit superscripts and subscripts when manipulating these functions, writing simply ext and int. Let us also remark that for Σ-types we could have also used p^* instead of $p_!$, making the evident changes. In the following, we shall use both, depending on which is more convenient.

Higher-dimensional categorical structure. Even if our development is entirely syntactic, many of the ideas presented in the paper are inspired by concepts of homotopy theory and higher-dimensional algebra. Therefore, we conclude this preliminary section by discussing some aspects of the relationship with higher-dimensional category theory, so as to provide further insight into our development.

First of all, observe that types and functions can be organized into an ordinary category, where the composition and identity laws hold as judgemental equalities. Indeed, if we define the composite $g \circ f : A \to C$ of $f : A \to B$ and $g : B \to C$ by letting

$$g \circ f =_{\mathrm{def}} (\lambda x : A) g(fx)\,,$$

and the identity $1_A : A \to A$ by letting $1_A =_{\mathrm{def}} (\lambda x : A)x$, the presence of the judgemental η-rule for Π-types in (1.3) in $\mathcal{H}$ implies that we have judgemental equalities

$$h \circ (g \circ f) = (h \circ g) \circ f\,, \quad 1_B \circ f = f\,, \quad f \circ 1_A = f\,. \tag{1.10}$$

Because of the strict associativity, we may omit bracketing of multiple composites and sometimes write simply gf instead of $g \circ f$.

The presence of identity types in our type theories, however, equips this category with additional structure. Each type A is a weak ∞-groupoid, having elements of A as objects, paths $p : \mathsf{Id}_A(a,b)$ as 1-morphisms (from a to b) and elements of iterated identity types as n-morphisms [23, 5]. We may write

$$q \cdot p : \mathsf{Id}_A(a,c)\,, \quad 1_a : \mathsf{Id}_A(a,a)\,, \quad p^{-1} : \mathsf{Id}_A(b,a)\,,$$

for the path obtained by composing $p : \mathsf{Id}_A(a,b)$ and $q : \mathsf{Id}_A(a,c)$, for the path $\mathsf{refl}(a) : \mathsf{Id}_A(a,a)$, and for the quasi-inverse of $p : \mathsf{Id}_A(a,b)$, respectively [16]. When manipulating this structure, we refer to the propositional equalities holding between various composites as the *groupoid laws*.

The category of types and functions can then be considered informally as enriched in ∞-groupoids (and hence as an $(\infty,1)$-category[1]), since function types $A \to B$, just like any other type, are ∞-groupoids. This $(\infty,1)$-category has types as objects, functions as 1-morphisms, paths $p : \mathsf{Id}_{A\to B}(f,g)$ as 2-morphisms, and higher paths as n-morphisms. We will not need all the structure of this higher-dimensional category (for which see [22]), but only some low-dimensional layers of it which can be defined easily. For example, given functions $f_1, f_2 : A \to B$, $g : B \to C$ and a path $p : \mathsf{Id}_{A\to B}(f_1, f_2)$, represented diagrammatically as

$$A \;\overset{f_1}{\underset{f_2}{\Downarrow p}}\; B \xrightarrow{\;g\;} C\,,$$

it is possible to define a path $g \circ p : \mathsf{Id}_{A\to C}(g \circ f_1\,, g \circ f_2)$.

Because of the equivalences $\mathsf{Id}(f,g) \simeq \mathsf{Hot}(f,g)$ recalled above, this $(\infty,1)$-category can be described equivalently as having types as objects, functions as 1-morphisms, homotopies $\alpha : \mathsf{Hot}(f,g)$ as 2-morphisms, and higher homotopies as n-morphisms. For example, given functions $f_1, f_2 : A \to B$, $g : B \to C$ and a homotopy $\alpha : \mathsf{Id}(f_1, f_2)$, there is a homotopy $g \circ \alpha : \mathsf{Hot}(g \circ f_1, g \circ f_2)$ which is defined so that, for every path $p : \mathsf{Id}_{A\to B}(f_1, f_2)$, the homotopies $\mathsf{ext}(g \circ p)$ and $g \circ \mathsf{ext}(p)$ are propositionally equal, where ext denotes the extension function for function types.

For the convenience of the reader, we summarize the different kinds of equalities (including logical equivalence) used in the paper in Table 5.

$A = B$	Definitional equality of types.
$A \simeq B$	Equivalence of types.
$A \leftrightarrow B$	Logical equivalence of types.
$a = b$	Definitional equality of elements.
$a \cong b$	Propositional equality of elements.
$f \sim g$	Homotopy of functions.

TABLE 5. Symbols for equality relations.

2. BIPOINTED TYPES

Bipointed types and bipointed morphisms. In this section and the next, we focus on the type Bool of Boolean truth values. Our development in these sections provides a template for what we will do for W-types in Section 4 and Section 5 and allows us to present the key ideas in a simpler context.

The rules for the type Bool that we consider here are given in Table 6. The introduction rules state that we have two canonical elements in Bool, written 0 and 1 here. The elimination rule can be understood as the propositions-as-types translation of an induction principle for Bool. Finally, the computation rules specify what happens if one applies the elimination rule immediately after applying the introduction rule.

Let us now suppose that we have a small type $A : \mathsf{U}$ and an equivalence $f : \mathsf{Bool} \to A$. Then, the type A has two distinguished elements $a_0 =_{\mathrm{def}} f(0)$ and $a_1 =_{\mathrm{def}} f(1)$, and it satisfies analogues of the elimination and computation rules for Bool, except that the conclusions of the computation

[1]We follow convention of using (∞,n)-category to denote an ∞-category in which k-morphisms, for $k > n$, are invertible. An ∞-groupoid is then the same thing as an $(\infty,0)$-category.

$$\mathsf{Bool} : \mathsf{type} \qquad 0 : \mathsf{Bool} \qquad 1 : \mathsf{Bool} \qquad \mathsf{Bool} : \mathsf{U}$$

$$\frac{x \in \mathsf{Bool} \vdash E(x) : \mathsf{type} \qquad e_0 : E(0) \qquad e_1 : E(1)}{x : \mathsf{Bool} \vdash \mathsf{boolelim}(x, e_0, e_1) : E(x)}$$

$$\frac{x \in \mathsf{Bool} \vdash E(x) : \mathsf{type} \qquad e_0 : E(0) \qquad e_1 : E(1)}{\mathsf{boolelim}(0, e_0, e_1) = e_0 : E(0)\,,}$$

$$\frac{x \in \mathsf{Bool} \vdash E(x) : \mathsf{type} \qquad e_0 : E(0) \qquad e_1 : E(1)}{\mathsf{boolelim}(1, e_0, e_1) = e_1 : E(1)}$$

TABLE 6. Rules for the type of Boolean truth values.

rules need to be modified by replacing the judgemental equalities with propositional ones. Our aim in this section is to provide a characterisation of the small types equivalent to Bool by means of a type-theoretical universal property. But in our development we do not need to assume to have the type Bool, and rather work in the type theory $\mathcal{H}$ specified in Section 1. We begin by introducing the notion of a bipointed type.

Definition 2.1. A *bipointed type* (A, a_0, a_1) is a type A equipped with two elements $a_0\,, a_1 : A$.

When referring to a bipointed type we sometimes suppress mention of its distinguished elements and write $A = (A, a_0, a_1)$ to recall this abuse of language. Similar conventions will be used throughout the paper for other kinds of structures. In the following, it will be convenient to represent a bipointed type A diagrammatically as follows:

$$1 \xrightarrow{a_0} A \xleftarrow{a_1} 1\,.$$

Here, the symbol 1 is purely a notational device, and does not represent the unit type, which is not assumed as part the type theory $\mathcal{H}$. The type Bool and its canonical elements $0, 1 : \mathsf{Bool}$ give us a bipointed type:

$$1 \xrightarrow{0} \mathsf{Bool} \xleftarrow{1} 1\,.$$

We say that a bipointed type $A = (A, a_0, a_1)$ is *small* if the type A is a small type, i.e. $A : \mathsf{U}$. Accordingly, the type of small bipointed types (which is not small) is then defined by letting

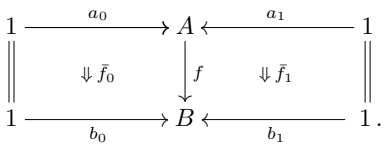
$$\mathsf{Bip} =_{\mathrm{def}} (\Sigma A : \mathsf{U})(A \times A)\,.$$

Next, we introduce the notion of a bipointed morphism between bipointed types. As one might imagine, a bipointed morphism consists of a function between the underlying types which preserves the bipointed structure. In our context, we formalize this by requiring the existence of appropriate paths, witnessing the preservation of structure, as the next definition makes precise.

Let us fix two bipointed types $A = (A, a_0, a_1)$ and $B = (B, b_0, b_1)$.

Definition 2.2. A *bipointed morphism* $(f, \bar{f}_0, \bar{f}_1) : A \to B$ is a function $f : A \to B$ equipped with paths $\bar{f}_0 : \mathsf{Id}(fa_0, b_0)$ and $\bar{f}_1 : \mathsf{Id}(fa_1, b_1)$.

Diagrammatically, we represent a bipointed morphism as follows:

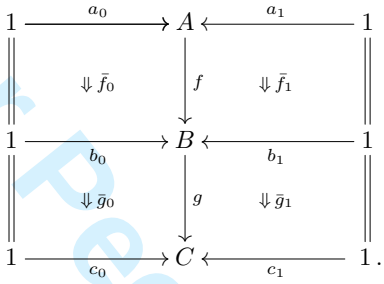


(2.1)

The type of bipointed morphisms from A to B is then defined by letting

$$\mathsf{Bip}(A,B) =_{\mathrm{def}} (\Sigma f : A \to B)\big(\mathsf{Id}(fa_0, b_0) \times \mathsf{Id}(fa_1, b_1)\big)\,.$$

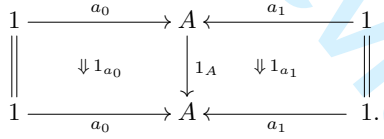
Bipointed types and their morphisms behave much like objects and morphisms in a category. Given two bipointed morphisms $(f, \bar{f}_0, \bar{f}_1) : A \to B$ and $(g, \bar{g}_0, \bar{g}_1) : B \to C$, we can define their composite as the triple consisting of the composite $gf : A \to C$ and the paths represented by the following pasting diagram:



Explicitly, for $k \in \{0,1\}$, the path $\overline{(gf)}_k : \mathsf{Id}(gfa_k, c_k)$ is obtained as the composite

$$gfa_k \xrightarrow{g \circ \bar{f}_k} gb_k \xrightarrow{\bar{g}_k} c_k\,.$$

Also, for any bipointed type $A = (A, a_0, a_1)$, the identity function $1_A : A \to A$ can be equipped with the structure of a bipointed by taking $\overline{(1_A)}_k : \mathsf{Id}(1_A(a_k), a_k)$ to be $1_{a_k} = \mathsf{refl}(a_k) : \mathsf{Id}(a_k, a_k)$ for $k \in \{0,1\}$. We represent this as the diagram



(2.2)

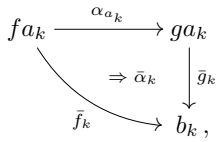
Note that, even if associativity and unit laws for composition for functions between types hold strictly (i.e. up to judgemental equality, cf. (1.10)), the associativity and unit laws for bipointed morphisms do not. This is due to the presence of paths in their definition, in complete analogy with the well-known situation in homotopy theory [9].

We have seen in Section 1 that for types A and B, the identity type of the function type $A \to B$ can be described equivalently as the type of homotopies between functions from A to B. As we show next, it is possible to extend this equivalence to bipointed morphisms. In order to do so, the next definition introduces the notion of a bipointed homotopy.

Let us now fix two bipointed morphisms $f = (f, \bar{f}_0, \bar{f}_1)$ and $g = (g, \bar{g}_0, \bar{g}_1)$ from A to B.

Definition 2.3. A *bipointed homotopy* $(\alpha, \bar{\alpha}_0, \bar{\alpha}_1) : f \to g$ is a homotopy $\alpha : \mathsf{Hot}(f, g)$ equipped with paths $\bar{\alpha}_0 : \mathsf{Id}(\bar{f}_0, \bar{g}_0 \cdot \alpha_{a_0})$ and $\bar{\alpha}_1 : \mathsf{Id}(\bar{f}_1, \bar{g}_1 \cdot \alpha_{a_1})$.

Diagrammatically, we represent the paths involved in a bipointed homotopy as follows:



for $k \in \{0,1\}$. The type of bipointed homotopies between f and g is then defined by letting

$$\mathsf{BipHot}\big((f,\bar{f}_0,\bar{f}_1),(g,\bar{g}_0,\bar{g}_1)\big) =_{\mathrm{def}} (\Sigma\alpha : \mathsf{Hot}(f,g))\big(\mathsf{Id}\big(\bar{f}_0,\bar{g}_0\cdot\alpha_{a_0}\big)\times\mathsf{Id}\big(\bar{f}_1,\bar{g}_1\cdot\alpha_{a_1}\big)\big)\,.$$

Lemma 2.4 essentially says that paths between bipointed morphisms are essentially the same thing as bipointed homotopies. This is the first instance of the suprising phenomenon, mentioned in the introduction, that identity types capture higher-dimensional algebraic structures in an apparently automatic way. It should also be pointed out that, as a consequence of the lemma, types of bipointed homotopies satisfy analogues of the rules for identity types.

Lemma 2.4. *The canonical function*

$$\mathsf{ext}^{\mathsf{Bip}}_{f,g} : \mathsf{Id}\big((f,\bar{f}_0,\bar{f}_1),(g,\bar{g}_0,\bar{g}_1)\big) \to \mathsf{BipHot}\big((f,\bar{f}_0,\bar{f}_1),(g,\bar{g}_0,\bar{g}_1)\big)\big)\,.$$

is an equivalence of types.

Proof. Recall that, for a path $p : \mathsf{Id}(f,g)$, we write $\mathsf{ext}\,p : \mathsf{Hot}(f,g)$ for the corresponding homotopy. We then have

$$\begin{aligned}
\mathsf{Id}\big((f,\bar{f}_0,\bar{f}_1),(g,\bar{g}_0,\bar{g}_1)\big) &\simeq (\Sigma p : \mathsf{Id}(f,g))\,\mathsf{Id}\big((\bar{f}_0,p^*(\bar{g}_0)\big)\times\mathsf{Id}\big(\bar{f}_1,p^*(\bar{g}_1)\big)\\
&\simeq (\Sigma p : \mathsf{Id}(f,g))\,\mathsf{Id}(\bar{f}_0,\bar{g}_0\cdot(\mathsf{ext}\,p)_{a_0})\times\mathsf{Id}(\bar{f}_1,\bar{g}_1\cdot(\mathsf{ext}\,p)_{a_1})\big)\\
&\simeq (\Sigma\alpha : \mathsf{Hot}(f,g))\,\mathsf{Id}(\bar{f}_0,\bar{g}_0\cdot\alpha_{a_0})\times\mathsf{Id}(\bar{f}_1,\bar{g}_1\cdot\alpha_{a_1})\\
&= \mathsf{BipHot}\big((f,\bar{f}_0,\bar{f}_1)\;(g,\bar{g}_0,\bar{g}_1)\big)\,,
\end{aligned}$$

as required. □

Fibered bipointed types and bipointed sections. Recall that for a dependent type

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

we referred to an element $f : (\Pi x : A)E(x)$ as a *section* of the dependent type. It will be convenient to extend this notion to bipointed types by introducing the following definition.

Let us fix a bipointed type $A = (A, a_0, a_1)$.

Definition 2.5. A *fibered bipointed type* (E, e_0, e_1) over A is a dependent type $x : A \vdash E(x) : \mathsf{type}$ equipped with elements $e_0 : E(a_0)$ and $e_1 : E(a_1)$.

The type of small fibered bipointed types over a bipointed type A is then defined by letting

$$\mathsf{FibBip}(A) =_{\mathrm{def}} (\Sigma E : A \to \mathsf{U})\big(E(a_0)\times E(a_1)\big)\,.$$

Let us now fix a fibered bipointed type $E = (E, e_0, e_1)$ over A.

The type $E' =_{\mathrm{def}} (\Sigma x : A)E(x)$ can be equipped with the structure of a a bipointed type by considering $e'_k =_{\mathrm{def}} \mathsf{pair}(a_k, e_k)$ (for $k \in \{0,1\}$) as distinguished elements of E'. In this way, the

first projection $\pi_1 : E' \to A$ becomes a bipointed morphism:

$$\begin{array}{ccccc} 1 & \xrightarrow{e'_0} & E' & \xleftarrow{e'_1} & 1 \\ \| & \overline{(\pi_1)_0} & \downarrow{\scriptstyle \pi_1} & \overline{(\pi_1)_1} & \| \\ 1 & \xrightarrow[a_0]{} & A & \xleftarrow[a_1]{} & 1\,. \end{array}$$

Definition 2.6. A *bipointed section* $(f, \bar{f}_0, \bar{f}_1)$ of E is a section $f : (\Pi x : A)E(x)$ equipped with paths $\bar{f}_0 : \mathsf{Id}_{E(a_0)}(fa_0, e_0)$ and $\bar{f}_1 : \mathsf{Id}_{E(a_1)}(fa_1, e_1)$.

The type of bipointed sections of E is then defined by letting

$$\mathsf{BipSec}(A, E) =_{\text{def}} (\Sigma f : (\Pi x : A)E(x)) \left(\mathsf{Id}_{E(a_0)}(fa_0, e_0) \times \mathsf{Id}_{E(a_1)}(fa_1, e_1)\right).$$

Given a bipointed section $f = (f, \bar{f}_0, \bar{f}_1)$ of E, we can define a bipointed morphism $f' : A \to E'$, where $E' = (E', e'_0, e'_1)$ is the bipointed type associated to E. Its underlying function is defined by $f' =_{\text{def}} (\lambda x : A)\mathsf{pair}(x, fx)$. With this definition, it is immediate to get the required paths $\bar{f'}_k : \mathsf{Id}(f'a_k, e'_k)$, for $k \in \{0, 1\}$. Note that the morphism $f' : A \to E'$ provides a right inverse for $\pi_1 : E' \to A$, since for every $x : A$ we have the judgemental equalities $\pi_1(f'x) = \pi_1\,\mathsf{pair}(x, fx) = x$. We represent this situation with the diagram

$$\begin{array}{ccc} & & E' \\ & {\scriptstyle \pi_1}\big\downarrow & \big)\,{\scriptstyle f'} \\ A. & & \end{array}$$

We characterize the identity type between two bipointed sections, using the notion of a bipointed homotopy. This is in complete analogy with what was done for bipointed morphisms in Lemma 2.4.

Let us now fix two bipointed sections $f = (f, \bar{f}_0, \bar{f}_1)$ and $g = (g, \bar{g}_0, \bar{g}_1)$ of E.

Definition 2.7. A *bipointed homotopy* $(\alpha, \bar{\alpha}_0, \bar{\alpha}_1) : f \to g$ is a homotopy $\alpha : \mathsf{Hot}(f, g)$ equipped with paths $\bar{\alpha}_0 : \mathsf{Id}(\bar{f}_0, \bar{g}_0 \cdot \alpha_{a_0})$ and $\bar{\alpha}_1 : \mathsf{Id}(\bar{f}_1, \bar{g}_1 \cdot \alpha_{a_1})$.

The type of bipointed homotopies between f and g as above is then defined by letting:

$$\mathsf{BipHot}\big((f, \bar{f}_0, \bar{f}_1), (g, \bar{g}_0, \bar{g}_1)\big) =_{\text{def}} (\Sigma \alpha : \mathsf{Hot}(f, g)) \left(\mathsf{Id}\big(\bar{f}_0, \bar{g}_0 \cdot \alpha_{a_0}\big) \times \mathsf{Id}\big(\bar{f}_1, \bar{g}_1 \cdot \alpha_{a_1}\big)\right).$$

Lemma 2.8. *The canonical function*

$$\mathsf{ext}^{\mathsf{BipHot}}_{f,g} : \mathsf{Id}\big((f, \bar{f}_0, \bar{f}_1), (g, \bar{g}_0, \bar{g}_1)\big) \to \mathsf{BipHot}\big((f, \bar{f}_0, \bar{f}_1), (g, \bar{g}_0, \bar{g}_1)\big)$$

is an equivalence of types.

Proof. The claim follows by an argument analogous to that of Lemma 2.4. □

Bipointed equivalences. We introduce the notion of equivalence between bipointed types and show in Proposition 2.12 that a bipointed morphism is an equivalence of bipointed types if and only if its underlying function is an equivalence of types. For this, we will use the characterization of equivalence of types as functions with a left and right inverse, which we recalled in Section 1. The characterization of bipointed equivalences given below will be used in Section 3 where we consider the counterpart of the univalence axiom for bipointed types.

Definition 2.9. We say that a bipointed morphism $f : A \to B$ is a *bipointed equivalence* if there exist bipointed morphisms $g : B \to A$ and $h : B \to A$ which provide a left and a right bipointed inverse for f, i.e. such that there exist paths $p : \mathsf{Id}_{\mathsf{Bip}(A,A)}(gf, 1_A)$ and $q : \mathsf{Id}_{\mathsf{Bip}(B,B)}(fh, 1_B)$.

For a bipointed morphism $f : A \to B$, the type of proofs that f is a bipointed equivalence is then defined by letting

$$\mathsf{isbipequiv}(f) =_{\mathrm{def}} (\Sigma g : \mathsf{Bip}(B, A))\, \mathsf{Id}_{\mathsf{Bip}(A,A)}(gf, 1_A) \times (\Sigma h : \mathsf{Bip}(A, B))\, \mathsf{Id}_{\mathsf{Bip}(B,B)}(fh, 1_B)\,,$$

and type of bipointed equivalences between A and B is defined by letting

$$\mathsf{BipEquiv}(A, B) =_{\mathrm{def}} (\Sigma f : \mathsf{Bip}(A, B))\, \mathsf{isbipequiv}(f)\,.$$

Since a bipointed equivalence is an equivalence with additional structure which ensures that it is well-behaved with respect to the bipointed structure, Lemma 2.10 below is essentially straightforward, but we include the details of the proof since we will need them to establish Proposition 2.12.

Lemma 2.10. *The underlying function of a bipointed equivalence is an equivalence of types. In particular, for every bipointed morphism $f : A \to B$ there is a function*

$$\pi_f : \mathsf{isbipequiv}(f) \to \mathsf{isequiv}(f)\,.$$

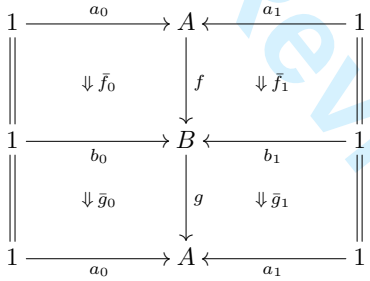
Proof. Let $f = (f, \bar{f}_0, \bar{f}_1)$ be a bipointed morphism from A to B. Unfolding the definition of $\mathsf{isbipequiv}(f)$ yields the type

$$(\Sigma g : B \to A)(\Sigma \bar{g}_0 : \mathsf{Id}(gb_0, a_0))(\Sigma \bar{g}_1 : \mathsf{Id}(gb_1, a_1))\, G(g, \bar{g}_0, \bar{g}_1) \times (\Sigma h : B \to A)(\Sigma \bar{h}_0 : \mathsf{Id}(hb_0, a_0))(\Sigma \bar{h}_1 : \mathsf{Id}(hb_1, a_1))\, H(h, \bar{h}_0, \bar{h}_1)\,, \quad (2.3)$$

where

$$G(g, \bar{g}_0, \bar{g}_1) =_{\mathrm{def}} \mathsf{Id}\big((gf\,, \overline{gf}_0\,, \overline{gf}_1)\,,\ (1_A\,, 1_{a_0}\,, 1_{a_1})\big)\,,$$
$$H(h, \bar{h}_0, \bar{h}_1) =_{\mathrm{def}} \mathsf{Id}\big((fh\,, \overline{fh}_0\,, \overline{fh}_1)\,,\ (1_B\,, 1_{b_0}\,, 1_{b_1})\big)\,.$$

The type $G(g, \bar{g}_0, \bar{g}_1)$ can be thought of as the type of proofs that the bipointed morphism $gf : A \to A$ is propositionally equal to the identity bipointed morphism $1_A : A \to A$, while $H(h, \bar{h}_0, \bar{h}_1)$ can be thought of as the type of proofs that the bipointed morphism $fh : B \to B$ is propositionally equal to the identity bipointed morphism $1_B : B \to B$. In particular, the elements of $G(g, \bar{g}_0, \bar{g}_1)$ can be thought of as proofs that the pasting diagram



is propositionally equal to the diagram in (2.2) representing the identity bipointed morphism.

Using the characterization of identity types of Σ-types in Section 1, the type $G(g, \bar{g}_0, \bar{g}_1)$ can be expressed equivalently as

$$(\Sigma p : \mathsf{Id}(gf, 1_A))\, \mathsf{Id}\big((\overline{gf}_0\,, \overline{gf}_1)\,,\ p^*(1_{a_0}\,, 1_{a_1})\big)\,,$$

where, for $p : \mathsf{Id}(gf, 1_A)$,

$$p^* : \big(\mathsf{Id}(1_A(a_0), a_0) \times \mathsf{Id}(1_A(a_1), a_1)\big) \to \big(\mathsf{Id}(gf(a_0), a_0) \times \mathsf{Id}(gf(a_1), a_1)\big)$$

is a transport function associated to p. Similarly, the type $H(h, \bar{h}_0, \bar{h}_1)$ is equivalent to

$$(\Sigma q : \mathsf{Id}(fh, 1_B))\, \mathsf{Id}\big((\overline{fh}_0\,, \overline{fh}_1)\,,\ q^*(1_{b_0}\,, 1_{b_1})\big)\,.$$

Thus, rearranging the order of the Σ-types in (2.3) and using the characterization of identity types in product types, we get that

$$\mathsf{isbipequiv}(f) \simeq (\Sigma g : B \to A)(\Sigma p : \mathsf{Id}(gf, 1_A))\, G'(g,p) \times (\Sigma h : B \to A)(\Sigma q : \mathsf{Id}(fh, 1_B))\, H'(h,q)\,, \quad (2.4)$$

where

$$G'(g,p) =_{\mathrm{def}} (\Sigma \bar{g}_0 : \mathsf{Id}(gb_0, a_0))\, \mathsf{Id}(\overline{gf}_0, p^*(1_{a_0})) \times (\Sigma \bar{g}_1 : \mathsf{Id}(gb_1, a_1))\, \mathsf{Id}(\overline{gf}_1, p^*(1_{a_1}))\,, \quad (2.5)$$

$$H'(h,q) =_{\mathrm{def}} (\Sigma \bar{h}_0 : \mathsf{Id}(hb_0, a_0))\, \mathsf{Id}(\overline{fh}_0, q^*(1_{b_0})) \times (\Sigma \bar{h}_1 : \mathsf{Id}(hb_1, a_1))\, \mathsf{Id}(\overline{fh}_1, q^*(1_{b_1}))\,. \quad (2.6)$$

Note that the elements of $G'(g,p)$ are 4-tuples consisting of paths $\bar{g}_0$, $\bar{g}_1$ making the function g into a bipointed morphism and of paths $\bar{p}_0$, $\bar{p}_1$ making the path p into a path between bipointed morphisms. Of course, the elements $H'(h,q)$ admits a similar description. The required function is then obtained by composing the equivalence in (2.4), the projection forgetting the components from $G'(g,p)$ and $H'(h,q)$, and the equivalence in (1.9). □

In Proposition 2.12 we will give an alternative characterisation of bipointed equivalences, which will be used in the proof of Theorem 3.14 and Corollary 3.7. Intuitively, it asserts that for every bipointed morphism $(f, \bar{f}_0, \bar{f}_1)$, if the underlying function f is an equivalence of types, there is an essentially unique way of making $(f, \bar{f}_0, \bar{f}_1)$ into a bipointed equivalence, i.e. of equipping the left and right inverses of f with the structure of bipointed morphisms so as to obtain bipointed inverses.[2] In order to prove this result, we need the following straightforward lemma.

Lemma 2.11.

(i) Let A be a type and $a, a_1, a_2 : A$. For paths $p_1 : \mathsf{Id}(a, a_1)$, $p_2 : \mathsf{Id}(a, a_2)$, the type

$$(\Sigma q : \mathsf{Id}_A(a_1, a_2))\, \mathsf{Id}(q \cdot p_1\,, p_2)$$

is contractible.

(ii) Let $f : A \to B$ be an equivalence, $a_1, a_2 : A$ and $b : B$. For paths $p_1 : \mathsf{Id}(b, fa_1)$, $p_2 : \mathsf{Id}(b, fa_2)$, the type

$$(\Sigma q : \mathsf{Id}_A(a_1, a_2))\, \mathsf{Id}((f \circ q) \cdot p_1, p_2)$$

is contractible.

Proposition 2.12. *A bipointed morphism $(f, \bar{f}_0, \bar{f}_1) : A \to B$ is a bipointed equivalence if and only if its underlying function $f : A \to B$ is an equivalence. In fact, the function*

$$\pi_f : \mathsf{isbipequiv}(f, \bar{f}_0, \bar{f}_1) \to \mathsf{isequiv}(f)\,.$$

is an equivalence of types.

Proof. Let $(f, \bar{f}_0, \bar{f}_1) : A \to B$ be a bipointed morphism. We wish to show that the homotopy fibers of the function π_f are contractible. So, let us fix a canonical element of $\mathsf{isequiv}(f)$, given by functions $g : B \to A$, $h : B \to A$ and paths $p : \mathsf{Id}(gf, 1_A)$ and $q : \mathsf{Id}(fh, 1_B)$. By the definition of π_f and standard facts about the homotopy fibers, we have an equivalence

$$\mathsf{hfiber}(\pi_f, (g, h, p, q)) \simeq G'(g,p) \times H'(h,q)\,,$$

where $G'(g,p)$ and $H'(h,q)$ are defined in (2.5) and (2.6), respectively. We claim that $G'(g,p)$ and $H'(h,q)$ are contractible. Since the proofs are essentially the same, we consider only $G'(g,p)$.

[2] This has several analogues in category theory. For example, consider monoidal categories $\mathbb{C}$ and $\mathbb{D}$ and a strong monoidal functor $F : \mathbb{C} \to \mathbb{D}$ which is an equivalence of categories. There is then an essentially unique way of making a quasi-inverse of F into a strong monoidal functor so as to obtain a monoidal equivalence.

Let $k \in \{0,1\}$. For a path $p : \mathsf{Id}(gf, 1_A)$, the path $p^*(1_{a_k}) : \mathsf{Id}(gfa_k, b_k)$ can be proved by Id-elimination to be propositionally equal to $(\mathsf{ext}\, p)_{a_k} : \mathsf{Id}(gfa_k, b_k)$, where $\mathsf{ext}\, p : \mathsf{Hot}(gf, 1_A)$. Combining this fact with the definition of composition of bipointed morphisms, we obtain that $G'(g,p)$ is equivalent to the product of the types

$$(\Sigma \bar{g}_k : \mathsf{Id}(gb_k, a_k))\, \mathsf{Id}(\bar{g}_k \cdot (g \circ \bar{f}_k), (\mathsf{ext}\, p)_{a_k})\,,$$

for $k \in \{0,1\}$, which are contractible by part (i) of Lemma 2.11. Hence $G'(h,p)$ is contractible, as required. □

Corollary 2.13. *For any bipointed morphism $(f, \bar{f}_0, \bar{f}_1)$, the type $\mathsf{isbipequiv}(f, \bar{f}_0, \bar{f}_1)$ is a mere proposition.* □

3. Homotopy-initial bipointed types

Inductive bipointed types. As we mentioned at the beginning of Section 2, if a type A is equivalent to Bool, then it satisfies the counterparts of the elimination and computation rules for Bool in which the computation rule is weakened by replacing the judgmental equality in its conclusion with a propositional equality. Using the notions of a fibered bipointed type and of a bipointed section introduced in Section 2, it is immediate to see that the these rules can be expressed equivalently by saying that every fibered bipointed type over A has a bipointed section (cf. [17]). Since bipointed types A of this kind play an important role in the following, we introduce some terminology[3] to refer to them.

Definition 3.1. A bipointed type A is said to be *inductive* if every small fibered bipointed type over it has a bipointed section, i.e. the type

$$\mathsf{isind}(A) =_{\mathrm{def}} (\Pi E : \mathsf{FibBip}(A))\mathsf{BipSec}(A, E)$$

is inhabited.

As we will see in Proposition 3.4, the type $\mathsf{isind}(A)$ is a mere proposition. We define the type of small inductive bipointed types by letting

$$\mathsf{BipInd} =_{\mathrm{def}} (\Sigma A : \mathsf{Bip})\mathsf{isind}(A)\,.$$

Thus, a canonical inductive bipointed type is given by a bipointed type $A = (A, a_0, a_1)$ together with a function which, given a fibered bipointed type $E = (E, e_0, e_1)$ over A, returns a bipointed section of E. Clearly, the type Bool is an inductive bipointed type. Furthermore, the property of being inductive can be transported along equivalences, in the sense that if A and B are equivalent bipointed types and A is inductive, then so is B. Thus, a type is equivalent to Bool if and only if it is inductive. Below, we begin to explore some consequences of the assumption that a bipointed type is inductive, with the goal of arriving at a characterisation of inductive bipointed types in Theorem 3.10.

Proposition 3.2. *Let $A = (A, a_0, a_1)$ be a bipointed type. Then A is inductive if only if we can derive rules of the form*

(i) the elimination rule

$$\frac{x : A \vdash E(x) : \mathsf{U} \qquad e_0 : E(a_0) \qquad e_1 : E(a_1)}{x : A \vdash \mathsf{elim}(x, e_0, e_1) : E(x)\,,}$$

[3] We use 'inductive' in analogy with the terminology used in set theory. This is not to be confused with the general notion of an inductive type.

(ii) the computation rules

$$\frac{x : A \vdash E(x) : \mathsf{U} \qquad e_0 : E(a_0) \qquad e_1 : E(a_1)}{\mathsf{comp}_k(e_0, e_1) : \mathsf{Id}\big(\mathsf{elim}(a_k, e_0, e_1), e_k\big)\,,}$$

where $k \in \{0, 1\}$.

Proof. Immediate. □

In the following, when we speak of an inductive bipointed type, we always assume that it comes equipped with functions elim and comp_k (for $k \in \{0, 1\}$) as in Proposition 3.2. Note that the rules in Proposition 3.2 are exactly the counterparts for A of the elimination rule and the weakening computation rules for Bool obtained by restricting the eliminating type to families of small dependent types[4] and, most importantly, replacing the judgemental equality in the conclusion with a propositional one, as mentioned above. The next proposition shows that, for an inductive bipointed type A, not only every fibered bipointed type over it has a section, but that such a section is unique up to a bipointed homotopy.

Proposition 3.3. *Let $A = (A, a_0, a_1)$ be a bipointed type. If A is inductive, then the following rules are derivable:*

(i) the η-rule

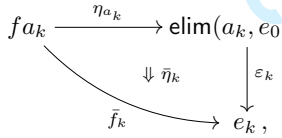
$$\frac{x : A \vdash E(x) : \mathsf{U} \quad e_0 : E(a_0) \quad e_1 : E(a_1) \quad x : A \vdash fx : E(x) \quad \bar{f}_0 : \mathsf{Id}(fa_0, e_0) \quad \bar{f}_1 : \mathsf{Id}(fa_1, e_1)}{x : A \vdash \eta_x : \mathsf{Id}(fx, \mathsf{elim}(x, e_0, e_1))}$$

(ii) the coherence rule

$$\frac{x : A \vdash E(x) : \mathsf{U} \quad e_0 : E(a_0) \quad e_1 : E(a_1) \quad x : A \vdash f(x) : E(x) \quad \bar{f}_0 : \mathsf{Id}(fa_0, e_0) \quad \bar{f}_1 : \mathsf{Id}(fa_1, e_1)}{\bar{\eta}_k : \mathsf{Id}\big(\mathsf{comp}_k(e_0, e_1) \cdot \eta_{a_k}\,,\ \bar{f}_k\big)}$$

where $k \in \{0, 1\}$.

Before proving Proposition 3.3, observe that the paths in the conclusion of the coherence rule can be represented diagrammatically in a way that is reminiscent of one of the triangular laws for an adjunction[5]:



where $\varepsilon_k =_{\mathrm{def}} \mathsf{comp}_k(e_0, e_1)$, for $k \in \{0, 1\}$.

Proof of Proposition 3.3. Let us assume the premisses of the η-rule. For $x : A$, define $F(x) : \mathsf{U}$ by letting $F(x) =_{\mathrm{def}} \mathsf{Id}_{E(x)}(fx, \mathsf{elim}(x, e_0, e_1))$. With this notation, proving the conclusion of the η-rule amounts to defining $\eta_x : F(x)$, for $x : A$. We do so using the elimination rule for A, as stated in Proposition 3.2. Thus, we need to find elements $p_k : F(a_k)$, for $k \in \{0, 1\}$. Since

$$F(a_k) = \mathsf{Id}(fa_k, \mathsf{elim}(a_k, e_0, e_1))\,,$$

we define p_k as the composite

$$fa_k \xrightarrow{\ \bar{f}_k\ } e_k \xrightarrow{\ \mathsf{comp}_k(e_0, e_1)^{-1}\ } \mathsf{elim}(a_k, e_0, e_1)\,.$$

[4] See Remark 3.13 for further discussion of this point.

[5] See Remark 3.9 for further discussion of this analogy.

For $x : A$, we can then defined the required element $\eta_x : F(x)$ by letting $\eta_x =_{\mathrm{def}} \mathsf{elim}(x, p_0, p_1)$. In order to prove the coherence rule, note that the computation rule of Proposition 3.2 gives us a path in $\mathsf{Id}(\eta_{a_k}, p_k)$, i.e. $\mathsf{Id}(\eta_{a_k}, \mathsf{comp}_k(e_0, e_1)^{-1} \cdot \bar{f}_k)$. The required paths can then be obtained using the groupoid laws. □

Proposition 3.4. *For every bipointed type $A = (A, a_0, a_1)$, the type $\mathsf{isind}(A)$ is a mere proposition.*

Proof. Recall that to prove that a type is a mere proposition, it suffices to do so under the assumption that it is inhabited. Assume therefore that $\mathsf{isind}(A)$ is inhabited. Since the dependent product of a family of mere propositions is again a mere proposition, it suffices to show that $\mathsf{BipSec}(A, E)$ is a mere proposition for any E. But for any two bipointed sections $f, g : \mathsf{BipSec}(A, E)$, there is a bipointed homotopy $\alpha : \mathsf{BipHot}(f, g)$ by Proposition 3.3 and hence, by Lemma 2.8, there is a path $p : \mathsf{Id}(f, g)$, as required. □

Homotopy-initial bipointed types. Let A be a small bipointed type and assume that it is inductive. We focus on the special case of fibered bipointed types that are constant, i.e. we have $E(x) = B$ for all $x : A$, where $B = (B, b_0, b_1)$ is a small bipointed type. Proposition 3.2 and Proposition 3.3 imply that there exists a bipointed morphism $f : A \to B$, which is unique in the sense that for any bipointed morphism $g : A \to B$ there is a bipointed homotopy $\alpha : \mathsf{BipHot}(f, g)$. Thus, by Lemma 2.4, there is a path $p : \mathsf{Id}_{\mathsf{Bip}(A,B)}(f, g)$. Furthermore, it can be shown that such a path is itself unique up to a higher path, which in turn is unique up to a yet higher path, and so on.

The key point in our development (described for Bool below and for W-types in Section 5) is that this sort of weak ∞-universality, which apparently involves infinitely much data, can be captured *fully within the system of type theory* (without resorting to coinduction) using ideas inspired by homotopy theory and higher-dimensional category theory. Indeed, in spite of the fact that bipointed types and morphisms do not form a category in a strict sense, it is possible to introduce the notion of a homotopy-initial bipointed type in completely elementary and explicit terms, as in Definition 3.5 below. This provides the template for the definition of a homotopy-initial algebra, which we will introduce in Section 5 in relation to W-types.

Definition 3.5. A small bipointed type A is said to be *homotopy-initial* if for any small bipointed type B, the type $\mathsf{Bip}(A, B)$ of bipointed morphisms from A to B is contractible, i.e. the type

$$\mathsf{ishinit}(A) =_{\mathrm{def}} (\Pi B : \mathsf{Bip})\, \mathsf{iscontr}(\mathsf{Bip}(A, B))$$

is inhabited.

Let us remark that the uniqueness implicit in Definition 3.5 requires that any two bipointed morphisms are propositionally equal as tuples. It should also be noted that the property of being homotopy-initial can be transported along equivalences, in the sense that if two bipointed types are equivalent, then one is homotopy-initial if and only if the other one is.

Proposition 3.6. *For every bipointed type A, the type $\mathsf{ishinit}(A)$ is a mere proposition.*

Proof. Recall that, for a type X, the type $\mathsf{iscontr}(X)$ is a mere proposition and that the dependent product of family of mere propositions is again a mere proposition. □

The next result is the counterpart of the familiar fact that objects characterized by universal properties are unique up to a unique isomorphism.

Proposition 3.7. *Homotopy-initial small bipointed types are unique up to a contractible type of bipointed equivalences, i.e. the type*

$$(\Pi A : \mathsf{Bip})(\Pi B : \mathsf{Bip})\big(\mathsf{ishinit}(A) \times \mathsf{ishinit}(B) \to \mathsf{iscontr}(\mathsf{BipEquiv}(A, B))\big)\,.$$

is inhabited.

Proof. Let A and B be homotopy-initial bipointed types. Recall that

$$\mathsf{BipEquiv}(A, B) = (\Sigma f : \mathsf{Bip}(A, B))\mathsf{isbipequiv}(f)\,.$$

We know that $\mathsf{Bip}(A, B)$ is contractible. Recalling that the dependent sum of a family of contractible types over a contractible type is contractible, it suffices to show that, for $f : \mathsf{Bip}(A, B)$, the type $\mathsf{isbipequiv}(f)$ is contractible. Since $\mathsf{isbipequiv}(f)$ is a mere proposition by Proposition 2.12, we only need to prove that it is inhabited. But the existence of a right and a left bipointed inverse for f follows immediately by the assumption that A and B are homotopy-initial. □

The next proposition spells out a characterization of homotopy-initial bipointed types in terms of type-theoretic rules.

Proposition 3.8. *A small bipointed type $A = (A, a_0, a_1)$ is homotopy-initial if and only if we can derive rules of the following form:*

(i) the recursion rule

$$\frac{B : \mathsf{U} \qquad b_0 : B \qquad b_1 : B}{x : A \vdash \mathsf{rec}(x, b_0, b_1) : B\,,}$$

(ii) the β-rules

$$\frac{B : \mathsf{U} \qquad b_0 : B \qquad b_1 : B}{\beta_k : \mathsf{Id}(\mathsf{rec}(a_k, b_0, b_1), b_k)\,,}$$

where $k \in \{0, 1\}$,

(iii) the η-rule

$$\frac{(B, b_0, b_1) : \mathsf{Bip} \quad (f, \bar{f}_0, \bar{f}_1) : \mathsf{Bip}(A, B)}{x : A \vdash \eta_x : \mathsf{Id}(fx, \mathsf{rec}(x, b_0, b_1))\,,}$$

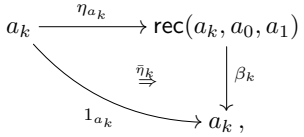
(iv) the (β, η)-coherence rule

$$\frac{(B, b_0, b_1) : \mathsf{Bip} \quad (f, \bar{f}_0, \bar{f}_1) : \mathsf{Bip}(A, B)}{\bar{\eta}_k : \mathsf{Id}(\beta_k \cdot \eta_{a_k}\,,\ \bar{f}_k)\,,}$$

where $k \in \{0, 1\}$.

Proof. The claim follows by unfolding the definition of homotopy-initiality. □

Remark 3.9. The terminology used for the rules in Proposition 3.8 is inspired by the special case that arises by considering B to be A itself and f to be the identity function. In this case, we obtain a function $(\lambda x : A)\mathsf{rec}(x, a_0, a_1) : A \to A$, paths $\beta_k : \mathsf{Id}(\mathsf{rec}(a_k, a_0, a_1), a_k)$, for $k \in \{0, 1\}$ and $\eta_x : \mathsf{Id}(x, \mathsf{rec}(x, a_0, a_1)$ and higher paths $\bar{\eta}_k$ fitting in the diagram



for $k \in \{0, 1\}$, which are analogous to one of the triangle laws for an adjunction.

The next theorem provides a characterisation of inductive bipointed types.

Theorem 3.10. *A small bipointed type A is inductive if and only if it is homotopy-initial, i.e. the type*

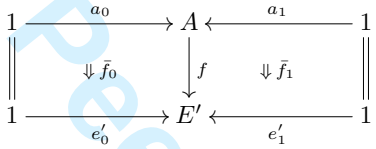
$$(\Pi A : \mathsf{Bip})\big(\mathsf{ishinit}(A) \leftrightarrow \mathsf{isind}(A)\big)$$

is inhabited

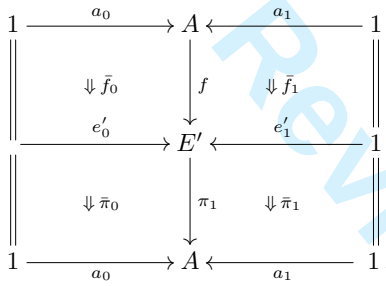
Proof. Let $A = (A, a_0, a_1)$ be a small bipointed type. We prove the two implications separately.

First, we show that if A is inductive then it is homotopy-initial. For this, it is sufficient to observe that the rules characterizing homotopy-initial bipointed types in Proposition 3.8 are special cases of the rules in Proposition 3.2 and Lemma 3.3, which are provable for inductive bipointed types.

Secondly, let us assume that $A = (A, a_0, a_1)$ is homotopy-initial and prove that it is inductive. For this, let $E = (E, e_0, e_1)$ be a fibered small bipointed type over A. We need to show that there exists a bipointed section $(s, \bar{s}_0, \bar{s}_1) : \mathsf{BipSec}(A, E)$. Let us consider the bipointed type associated to E, with carrier $E' =_{\mathrm{def}} (\Sigma x : A)E(x)$ and distinguished elements $e'_k =_{\mathrm{def}} \mathsf{pair}(a_k, e_k)$, for $k \in \{0, 1\}$. In this way, the first projection $\pi_1 : E' \to A$ is a bipointed morphism. By the homotopy-initiality of A, we have a bipointed morphism $(f, \bar{f}_0, \bar{f}_1) : (A, a_0, a_1) \to (E', e'_0, e'_1)$, which we represent with the diagram



We can compose $f : A \to E'$ with $\pi_1 : E' \to A$ and obtain a bipointed morphism $\pi_1 f : A \to A$, which is represented by the diagram



Since the identity $1_A : A \to A$ is also a bipointed morphism, by the homotopy-initiality of A there is an element of $\mathsf{Id}_{\mathsf{Bip}(A,A)}(\pi_1 f, 1_A)$. By Lemma 2.4, this gives us a bipointed homotopy $(\alpha, \bar{\alpha}_0, \bar{\alpha}_1) : \mathsf{BipHot}(\pi_1 f, 1_A)$. This amounts to a homotopy $\alpha : \mathsf{Hot}(\pi_1 f, 1_A)$ and paths

$$\bar{\alpha}_k : \mathsf{Id}(\overline{(\pi_1 f)}_k\,,\ \alpha_{a_k} \cdot 1_{a_k})\,,$$

for $k \in \{0, 1\}$. We begin to define the required bipointed section by defining, for $x : A$,

$$s(x) =_{\mathrm{def}} (\alpha_x)_!\big(\pi_2 f x\big)\,,$$

where $(\alpha_x)_! : E(\pi_1 f x) \to E(x)$. We now construct paths $\bar{s}_k : \mathsf{Id}(sa_k, e_k)$, for $k \in \{0, 1\}$. First of all, recall that $\bar{f}_k : \mathsf{Id}(f a_k, e'_k)$, where $e'_k = \mathsf{pair}(a_k, e_k) : (\Sigma x : A)E(x)$. Using the characterization of identity types of Σ-types, we define

$$p =_{\mathrm{def}} \pi_1\, \mathsf{ext}^{\Sigma}(\bar{f}_k) : \mathsf{Id}(\pi_1 f a_k\,,\ \pi_1 e'_k)\,, \quad q =_{\mathrm{def}} \pi_2\, \mathsf{ext}^{\Sigma}(\bar{f}_k) : \mathsf{Id}(p_!(\pi_2 f a_k), \pi_2 e'_k)\,.$$

Now, note that

$$\mathsf{Id}_A(\pi_1 f a_k\,,\ \pi_1 e'_k) = \mathsf{Id}_A(\pi_1 f a_k, a_k)\,, \quad \mathsf{Id}_{E(a_k)}(p_!(\pi_2 f a_k), \pi_2 e'_k) = \mathsf{Id}_{E(a_k)}(s a_k, e_k)$$

and that we have

$$\begin{aligned}
\overline{(\pi_1 f)}_k &\cong (\overline{\pi_1})_k \cdot (\pi_1 \circ \bar{f}_k) && \text{(by definition of } \pi_1 f) \\
&\cong 1_{a_k} \cdot (\pi_1 \circ \bar{f}_k) && \text{(by definition of } \pi_1) \\
&\cong (\pi_1 \circ \bar{f}_k) && \text{(by the groupoid laws)} \\
&\cong p && \text{(by definition of } \mathsf{ext}^\Sigma)\,.
\end{aligned}$$

Therefore, we can construct the following chain of paths:

$$\begin{aligned}
p &\cong \overline{(\pi_1 f)}_k && \text{(by what we just proved)} \\
&\cong 1_{a_k} \cdot \alpha_{a_k} && \text{(by the path } \bar{\alpha}_k) \\
&\cong \alpha_{a_k} && \text{(by the groupoid laws)}
\end{aligned}$$

Hence, the required path $\bar{s}_k : \mathsf{Id}(s a_k, e_k)$ can be defined as the following composite:

$$\begin{aligned}
s a_k &= (\alpha_{a_k})_!\big(\pi_2 f a_k\big) && \text{(by the definition of } s) \\
&\cong p_!\big(\pi_2 f a_k\big) && \text{(since } p \cong \alpha_{a_k}) \\
&\cong e_k && \text{(by the path } q)\,.
\end{aligned}$$

This concludes the proof. □

The proof of Theorem 3.10 simplifies considerably within the extensional type theory $\mathcal{H}^{\mathrm{ext}}$ obtained by adding to $\mathcal{H}$ the identity reflection rule in (1.2). In that type theory, there is a judgemental equality between the composite $\pi_1 f : A \to A$ and the identity $1_A : A \to A$, with which the rest of the argument can be shortened considerably. In that setting, one obtains the familiar characterisation of an inductive type as strict initial algebras.

Theorem 3.10 gives a logical equivalence between two types, but in fact we have a genuine equivalence of types, as the following corollary shows.

Corollary 3.11. *For a bipointed type A, there is an equivalence of types* $\mathsf{isind}(A) \simeq \mathsf{ishinit}(A)$.

Proof. Theorem 3.10 gives a logical equivalence, but $\mathsf{isind}(A)$ is a mere proposition by Proposition 3.4 and $\mathsf{ishinit}(A)$ is a mere proposition by Proposition 3.6. □

The next proposition characterizes the type Bool up to equivalence. In its statement, we refer to the rules for Bool in Table 6.

Corollary 3.12. *Assuming the rules for the type* Bool*, for a bipointed type $A = (A, a_0, a_1)$, the following conditions are equivalent:*

(i) A is inductive,
(ii) A is homotopy-initial,
(iii) A and Bool *are equivalent as bipointed types.*

In particular, Bool *is a homotopy-initial bipointed type.* □

Remark 3.13. Note that the elimination rules for Bool allow us to eliminate over an arbitrary, i.e. not necessarily small, dependent type. Instead, the definition of an inductive bipointed type involve the existence of sections over small fibered bipointed types. In spite of this apparent difference, since Bool is assumed to be a small type, one can prove an equivalence between any

inductive type A and Bool and hence derive counterparts of the elimination rules for Bool for any inductive bipointed type.

Let us point out that there are at least two alternatives to the approach taken here regarding universes. The first involves avoiding the restriction to *small* fibered bipointed types in the definition of the notion of an inductive bipointed type. Accordingly, one drops the restriction of mapping into *small* bipointed types in the definition of a notion of a homotopy-initial algebra. With these changes, there is still a logical equivalence between the modified notions, but this is no longer an internal statement in the type theory, as in Theorem 3.10. Alternatively, one could assume to have a hiearchy of type universes $\mathsf{U}_0 : \mathsf{U}_1 : \ \ldots \ : \mathsf{U}_n : \mathsf{U}_{n+1} : \ldots$ and modify the elimination rules for Bool by specifying that the types into which we are eliminating belong to some universe. A counterpart of Theorem 3.10, now stated with appropriate universe levels, would still hold.

Univalence for bipointed types. We conclude this section by showing that if the type universe U is assumed to be univalent, then a form of the univalence axiom holds also for bipointed types, in the sense made precise by the next theorem, where we use notation analogous to the one introduced for extension functions in Section 1. This is an instance of the Structure Identity Principle considered in [2].

Theorem 3.14. *Assuming the univalence axiom, for small bipointed types $A, B : \mathsf{Bip}$, the canonical function*

$$\mathsf{ext}^{\mathsf{Bip}}_{A,B} : \mathsf{Id}_{\mathsf{Bip}}\big(A, B\big) \to \mathsf{BipEquiv}(A, B)$$

is an equivalence.

Proof. Let $(A, a_0, a_1), (B, b_0, b_1)$ be small bipointed types. By the characterization of the identity types of Σ-types, the identity type $\mathsf{Id}\big((A, a_0, a_1), (B, b_0, b_1)\big)$ is equivalent to the type

$$(\Sigma p : \mathsf{Id}_{\mathsf{U}}(A, B))\, \mathsf{Id}((a_0, a_1), p^*(b_0, b_1))\,.$$

By Id-elimination and the characterization of paths in product types, this type is equivalent to

$$(\Sigma p : \mathsf{Id}_{\mathsf{U}}(A, B))\, \mathsf{Id}\big((\mathsf{ext}\, p)(a_0), b_0\big) \times \mathsf{Id}\big((\mathsf{ext}\, p)(a_1), b_1\big)\,,$$

where $\mathsf{ext}\, p : A \to B$ is the equivalence of types associated to $p : \mathsf{Id}_{\mathsf{U}}(A, B)$. By the univalence axiom, the above type is equivalent to

$$(\Sigma f : \mathsf{Equiv}(A, B))\, \mathsf{Id}\big(fa_0, b_0\big) \times \mathsf{Id}\big(fa_1, b_1\big)\,.$$

After rearranging, we get

$$(\Sigma f : A \to B)(\Sigma \bar{f}_0 : \mathsf{Id}(fa_0, b_0))(\Sigma \bar{f}_1 : \mathsf{Id}(fa_1, b_1))\, \mathsf{isequiv}(f)\,,$$

which is equivalent to $\mathsf{BipEquiv}(A, B)$ by Proposition 2.12. Finally, it is not hard to see that the composition of the above equivalences yields the function $\mathsf{ext}^{\mathsf{Bip}}_{A,B}$ up to a homotopy, thus showing that it is an equivalence, as required. □

Corollary 3.15. *Assuming the univalence axiom, homotopy-initial small bipointed types are unique up to a contractible type of paths, i.e. the type*

$$(\Pi A : \mathsf{Bip})(\Pi B : \mathsf{Bip})\big(\mathsf{ishinit}(A) \times \mathsf{ishinit}(B) \to \mathsf{iscontr}(\mathsf{Id}_{\mathsf{Bip}}(A, B))\big)\,.$$

is inhabited.

Proof. This is an immediate consequence of Proposition 3.7 and Theorem 3.14. □

4. Polynomial functors and their algebras

Algebras and algebra morphisms. The main aim of this paper is to carry out an analysis for well-ordering types (introduced in [25]), or W-types, analogous to the one we have just done for the type Bool. We recall the rules for W-types in Table 7. There, we sometimes write W for $(\mathrm{W}x : A)B(x)$ for brevity. Informally, a W-type can be seen as the free algebra for a signature with arbitrarily many operations of possibly infinite arity, but no equations. The premises of the formation rule can be thought of as specifying a signature that has the elements of the type A as (names of) 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 usual, the computation rule states what happens if we apply the the elimination rule to a canonical element of the inductive type. Finally, we have a rule expressing the closure of the type universe U under the formation of W-types.

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

$$\frac{w : W \vdash E(w) : \mathsf{type} \quad x : A,\ u : B(x) \to W,\ v : (\Pi y : B(x))E(uy) \vdash e(x, u, v) : E(\mathsf{sup}(x, u))}{w : W \vdash \mathsf{elim}(w, e) : E(w)}$$

$$\frac{w : W \vdash E(w) : \mathsf{type} \quad x : A,\ u : B(x) \to W,\ v : (\Pi y : B(x))E(uy) \vdash e(x, u, v) : E(\mathsf{sup}(x, u))}{x : A,\ u : B(x) \to W \vdash \mathsf{elim}(\mathsf{sup}(x, u), e) = e(x, u, (\lambda y : B(x))\,\mathsf{elim}(uy, e)) : E(\mathsf{sup}(x, u))}$$

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

Table 7. Rules for W-types.

We now consider a small type $A : \mathsf{U}$ and a small dependent type $B : A \to \mathsf{U}$, which we consider fixed for this section and the next. For $C : \mathsf{U}$, we define

$$PC =_{\mathrm{def}} (\Sigma x : A)(B(x) \to C)\,.$$

In this way, we obtain a function $P : \mathsf{U} \to \mathsf{U}$. This operation on types extends to an operation on functions, as follows. For $f : C \to D$, we define $Pf : PC \to PD$ by Σ-elimination so that, for $x : A$ and $u : B(x) \to C$, we have

$$(Pf)((x, u)) = (x, fu)\,.$$

This assignment is pseudo-functorial in the sense that we have propositional, rather than judgemental, equalities:

$$\phi_{f,g} : \mathsf{Id}(P(g \circ f), Pg \circ Pf)\,, \quad \phi_A : \mathsf{Id}(P(1_A), 1_{PA}) \tag{4.1}$$

for $f : C \to D$, $g : D \to E$. We still refer to P as the *polynomial functor* associated to $A : \mathsf{U}$ and $B : A \to \mathsf{U}$, so as to highlight the analogy with the theory of polynomial functors on locally cartesian closed categories [13, 27]. We shall also use that P acts on homotopies: given functions $f, g : C \to D$ and a homotopy $\alpha : \mathsf{Hot}(f, g)$, it is possible to define a homotopy $P\alpha : \mathsf{Hot}(Pf, Pg)$. Explicitly, for $x : A$ and $u : B(x) \to C$, we define $(P\alpha)_{x,u} : \mathsf{Id}((x, fu), (x, gu))$ by letting

$$(P\alpha)_{x,u} =_{\mathrm{def}} (x, \mathsf{int}(\alpha_u))\,, \tag{4.2}$$

where α_u is the evident homotopy between fu and gu, and int is the function transforming homotopies into paths discussed in Section 1.

Definition 4.1. A *P-algebra*

$$(C, \mathrm{sup}_C)$$

is a small type $C : \mathsf{U}$ equipped with a function $\mathrm{sup}_C : PC \to C$.

The type of P-algebras is then defined as

$$\mathsf{Alg} =_{\mathrm{def}} (\Sigma C : \mathsf{U})(PC \to C)\,.$$

Given a P-algebra $C = (C, \mathrm{sup}_C)$, we refer to the type C as the *carrier* or *underlying type* of the algebra and to the function $\mathrm{sup}_C : PC \to C$ as the *structure map* of the P-algebra. In the presence of W-types, an example of P-algebra is given by the type $W =_{\mathrm{def}} (\mathrm{W}x : A)B(x)$, with structure map given by the introduction rule for W-types.

Let us now fix P-algebras $C = (C, \mathrm{sup}_C)$ and $D = (D, \mathrm{sup}_D)$.

Definition 4.2. A *P-algebra morphism* $(f, \bar{f}) : C \to D$ is a function $f : C \to D$ equipped with a path $\bar{f} : \mathsf{Id}(f \circ \mathrm{sup}_C\,, \mathrm{sup}_D \circ Pf)$.

Note that the homotopy associated to a path $\bar{f}$ as above has components

$$(\mathsf{ext}\,\bar{f})_{x,u} : \mathsf{Id}\big(f(\mathrm{sup}_C(x,u)), \mathrm{sup}_D(x, fu)\big)\,.$$

A P-algebra morphism as above can be represented with a diagram of the form

$$\begin{array}{ccc} PC & \xrightarrow{\mathrm{sup}_C} & C \\ {\scriptstyle Pf}\downarrow & \Downarrow \bar{f} & \downarrow{\scriptstyle f} \\ PD & \xrightarrow[\mathrm{sup}_D]{} & D\,. \end{array}$$

We use this slighly unconventional orientation of the diagram in order to stress the analogy with bipointed morphisms (cf. the diagram in (2.1)). Informally, one can think of the path $\bar{f}$ as a proof that the diagram commutes (which is the requirement defining the notion of morphism of endofunctor algebras in category theory) or as an invertible 2-cell (as in the notion of a pseudo-morphism between algebras in 2-dimensional category theory [8]). For later use, let us introduce some auxiliary notation. For P-algebras $C = (C, \mathrm{sup}_C)$, $D = (D, \mathrm{sup}_D)$ and a function $f : C \to D$ between their underlying types, let us define

$$\mathsf{isalghom}(f) =_{\mathrm{def}} \mathsf{Id}(f \circ \mathrm{sup}_C\,, \mathrm{sup}_D \circ Pf)\,. \tag{4.3}$$

Note that this type is not, in general, a mere proposition. Informally, $\mathsf{isalghom}(f)$ is the type of paths $\bar{f}$ witnessing that f is a P-algebra morphism, fitting in a diagram as above. Accordingly, the type of P-algebra morphisms between C and D is defined by

$$\mathsf{Alg}(C, D) =_{\mathrm{def}} (\Sigma f : C \to D)\,\mathsf{isalghom}(f)\,.$$

We now define the composition operation for P-algebra morphisms. Given $(f, \bar{f}) : C \to D$ and $(g, \bar{g}) : D \to E$, their composite $(gf, \overline{gf}) : (C, \mathrm{sup}_C) \to (E, \mathrm{sup}_E)$ is obtained as follows. Its underlying function is given by $gf : C \to E$, and so the the required path must be of the form

$$\overline{(gf)} : \mathsf{Id}\big((g \circ f) \circ \mathrm{sup}_C\,, \mathrm{sup}_E \circ P(g \circ f)\big)\,.$$

Such a path is obtained by pasting the diagrams

$$\begin{array}{ccc} PC & \xrightarrow{\sup_C} & C \\ {\scriptstyle Pf}\downarrow & \Downarrow \bar{f} & \downarrow{\scriptstyle f} \\ PD & \xrightarrow[\sup_D]{} & D \\ {\scriptstyle Pg}\downarrow & \Downarrow \bar{g} & \downarrow{\scriptstyle g} \\ PE & \xrightarrow[\sup_E]{} & E\,. \end{array}$$

More precisely, it is given by the following composition of paths:

$$g \circ f \circ \sup_C \xrightarrow{g \circ \bar{f}} g \circ \sup_D \circ Pf \xrightarrow{\bar{g} \circ Pf} \sup_E \circ Pg \circ Pf \xrightarrow{\sup_E \circ \phi^{-1}_{f,g}} \sup_E \circ P(g \circ f)\,,$$

where we used the pseudo-functoriality of P in (4.1). For a P-algebra C, the identity function $1_C : C \to C$ has an evident structure of P-algebra morphism, represented in the diagram

$$\begin{array}{ccc} PC & \xrightarrow{\sup_C} & C \\ {\scriptstyle P(1_C)}\downarrow & \Downarrow \bar{1}_C & \downarrow{\scriptstyle 1_C} \\ PC & \xrightarrow[\sup_C]{} & C\,. \end{array} \tag{4.4}$$

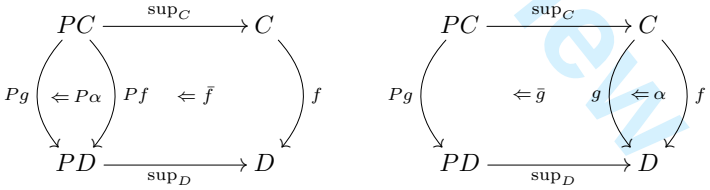
As in the case of bipointed types, the associativity and unit laws for a category do not hold up to judgemental equality, but only so up to a system of higher and higher paths.

We will require an alternative description of the identity type between two P-algebra morphisms. For this, we introduce the notion of a P-algebra homotopy in the next definition.

Let us fix P-algebra morphisms $f = (f, \bar{f})$ and $g = (g, \bar{g})$ from C to D. The next definition makes use of the action of P on homotopies defined in (4.2).

Definition 4.3. A *P-algebra homotopy* $(\alpha, \bar{\alpha}) : f \to g$ is a homotopy $\alpha : \mathsf{Hot}(f, g)$ equipped with a homotopy $\bar{\alpha} : \mathsf{Hot}\big((\sup_D \circ P\alpha) \cdot (\mathsf{ext}\, \bar{f})\,,\ (\mathsf{ext}\, \bar{g}) \cdot (\alpha \circ \sup_C)\big)$.

Note that in the definition $\alpha \circ \sup_C$ and $\sup_D \circ P\alpha$ are obtained by pre-composition and post-compositions, respectively, of functions with homotopies. The homotopy $\bar{\alpha}$ can be thought of as a proof that the two homotopies produced by the pasting diagrams



are equal, which is analogous to the condition defining an algebra 2-cell in 2-dimensional category theory [8]. Explicitly, the component of $\bar{\alpha}$ associated to $x : A$ and $u : B(x) \to C$ fits into diagrams of the form

$$\begin{array}{ccc} f(\sup_C(x,u)) & \xrightarrow{(\mathsf{ext}\bar{f})_{x,u}} & \sup_D(x, fu) \\ {\scriptstyle \alpha_{\sup_C(x,u)}}\downarrow & \Downarrow \bar{\alpha}_{x,u} & \downarrow{\scriptstyle \sup_D(x, \mathsf{int}(\alpha_u))} \\ g(\sup_C(x,u)) & \xrightarrow[(\mathsf{ext}\bar{g})_{x,u}]{} & \sup_D(x, gu)\,, \end{array}$$

where $\mathsf{int}(\alpha_u)$ denotes the path associated to the homotopy $(\lambda y : B(x))\alpha_{uy}$ between fu and gu. The type of P-algebra homotopies is then defined by

$$\mathsf{AlgHot}\big((f,\bar{f}),(g,\bar{g})\big) =_{\mathrm{def}} (\Sigma\alpha : \mathsf{Hot}(f,g))\, \mathsf{Hot}\big((\mathrm{sup}_D \circ P\alpha)\cdot(\mathsf{ext}\,\bar{f})\,,\ ((\mathsf{ext}\,\bar{g}))\cdot(\alpha \circ \mathrm{sup}_C)\big)\,.$$

Lemma 4.4. *For every pair of P-algebra morphisms $(f,\bar{f}),(g,\bar{g}) : C \to D$, the canonical function*

$$\mathsf{ext}^{\mathsf{Alg}}_{f,g} : \mathsf{Id}\big((f,\bar{f}),(g,\bar{g})\big) \to \mathsf{HotAlg}\big((f,\bar{f}),(g,\bar{g})\big).$$

is an equivalence of types.

Proof. This follows from a more general statement to be proved in Lemma 4.8 below. □

This is another case of the identity type encoding higher-categorical structure; we note that the proof of Lemma 4.4 does not require the univalence axiom.

Fibered algebras and algebra sections. We now introduce the fibered versions of the notions of a P-algebra, P-algebra morphism, and P-algebra homotopy. Some preliminary remarks will help us to motivate our definitions. Let us consider a fixed P-algebra $C = (C, \mathrm{sup}_C)$. Given a dependent type $E : C \to \mathsf{U}$, we wish to describe what data determines a P-algebra structure on the type $E' =_{\mathrm{def}} (\Sigma z : C)E(z)$. First of all, using a special case of the $\Pi\Sigma$-distributivity law recalled in (1.8), we have

$$PE' \simeq (\Sigma x : A)(\Sigma u : B(x) \to C)(\Pi y : B(x))E(uy)\,.$$

Therefore, we obtain

$$\begin{aligned} PE' \to E' &\simeq \Big((\Sigma x : A)(\Sigma u : B(x) \to C)(\Pi y : B(x))E(uy)\Big) \to (\Sigma z : C)E(z) \\ &\simeq (\Pi x : A)(\Pi u : B(x) \to C)(\Pi v : (\Pi y : B(x))E(uy))(\Sigma z : C)E(z)\,, \end{aligned}$$

and so a structure map $\mathrm{sup}_{E'} : PE' \to E'$ can be viewed equivalently as a function which takes arguments $x : A$, $u : B(x) \to C$, $v : (\Pi y : B(x))E(uy)$ and returns an element of E'. Thus, if we wish to ensure that the structure map $\mathrm{sup}_{E'} : PE' \to E'$ is such that the projection function $\pi_1 : E' \to C$ is a P-algebra morphism, i.e. that we can find a path fitting in the diagram

$$\begin{array}{ccc} PE' & \xrightarrow{\mathrm{sup}_{E'}} & E' \\ {\scriptstyle P\pi_1}\downarrow & \Downarrow \overline{\pi}_1 & \downarrow{\scriptstyle \pi_1} \\ PC & \xrightarrow[\mathrm{sup}_C]{} & C\,, \end{array}$$

it is sufficient to require the existence of a function of the form

$$e : (\Pi x : A)(\Pi u : B(x) \to C)(\Pi v : (\Pi y : B(x))E(uy))E(\mathrm{sup}_C(x,u))\,.$$

Note that such a function appears also in one of the premisses of the elimination rule for W-types in Table 7. We are therefore led to make the following definition.

Definition 4.5. A *fibered P-algebra* over C consists of a dependent type $E : C \to \mathsf{U}$ and a function $e : (\Pi x : A)(\Pi u : B(x) \to C)((\Pi y : B(x))E(uy))\, E(\mathrm{sup}_C(x,u))$.

We define the type of fibered P-algebras over C as follows:

$$\mathsf{FibAlg}(C) =_{\mathrm{def}} (\Sigma E : C \to \mathsf{U})(\Pi x : A)(\Pi u : B(x) \to C)((\Pi y : B(x))E(uy))\, E(\mathrm{sup}_C(x,u))$$

Let us consider a fixed fibered P-algebra $E = (E, e)$ over C.

We define the P-algebra $E' = (E', \sup_{E'})$, to which we shall refer as the *P-algebra associated to* E, as follows. As before, we define $E' =_{\text{def}} (\Sigma z : C)E(z)$ and $\sup_{E'} : PE' \to E'$ by Σ-elimination so that, for $x : A$ and $u : B(x) \to E'$, we have

$$\sup_{E'}(x, u) = \mathsf{pair}\big(\sup_C(x, \pi_1 u)\,, e(x, \pi_1 u, \pi_2 u)\big)\,.$$

Here, note that $\pi_1 u : B(x) \to C$ and $\pi_2 u : (\Pi y : B(x))E(\pi_1 uy)$ and so, by the type of e, we have that $e(x, \pi_1 u, \pi_2 u) : E(\sup_C(x, \pi_1 u))$, as required.

In analogy with the way we defined fibered P-algebras, it is possible to define P-algebra sections. To state this definition, we need some preliminary notation. For $f : (\Pi z : C)E(z)$, we define $e_f : (\Pi v : PC)E(\sup_C(v))$ so that, for $x : A$, $u : B(x) \to C$, we have

$$e_f(x, u) =_{\text{def}} e(x, u, fu)\,. \tag{4.5}$$

Here, note that for $y : B(x)$, we have $uy : C$ and hence $fuy : E(uy)$, as required.

Definition 4.6. A *P-algebra section* $(f, \bar{f})$ of E is a section $f : (\Pi z : C)E(z)$ equipped with a path $\bar{f} : \mathsf{Id}\big(f \sup_C\,, e_f\big)$.

Note that the components of the homotopy $\mathsf{ext}\,\bar{f}$ associated to a path $\bar{f}$ as above have the form $(\mathsf{ext}\,\bar{f})_{x,u} : \mathsf{Id}\big(f(\sup_C(x, u))\,, e(x, u, fu)\big)$. We define the type of P-algebra sections of E by letting

$$\mathsf{AlgSec}(C, E) =_{\text{def}} (\Sigma f : (\Pi x : C)E(x))\ \mathsf{Id}\big(f \sup_C\,, e_f\big)\,.$$

This terminology is justified by the fact that, given an algebra section $(f, \bar{f})$ of E, there is an algebra morphism $f' : C \to E'$, where $E' = (E', \sup_{E'})$ is the P-algebra associated to E. Its underlying function is defined by letting $f' =_{\text{def}} (\lambda z : C)\,\mathsf{pair}(z, fz)$ and direct calculations show that there is a path fitting in the diagram

$$\begin{array}{ccc} PC & \xrightarrow{\sup_C} & C \\ {\scriptstyle Pf'}\downarrow & \Downarrow \overline{f'} & \downarrow{\scriptstyle f'} \\ PE' & \xrightarrow[\sup_{E'}]{} & E'\,. \end{array}$$

This P-algebra morphism provides a section of the P-algebra morphism $\pi_1 : E' \to C$ in the sense that the composite P-algebra morphism $\pi_1 f' : C \to C$ can be shown to be propositionally equal to the identity P-algebra morphism $1_C : C \to C$.

We will require an analysis of paths between of P-algebra sections and thus we introduce, in Definition 4.7 below, the notion of a homotopy between P-algebra sections. In order to state the definition more briefly, let us introduce some notation. For a fibered P-algebra $E = (E, e)$, sections $f, g : (\Pi z : C)E(z)$ and a path $p : \mathsf{Id}(f, g)$, we write $e_p : \mathsf{Id}(e_f, e_g)$ for the evident path defined by Id-elimination, where e_f and e_g are defined as in (4.5). By the characterisation of identity types of function types, a homotopy $\alpha : \mathsf{Hot}(f, g)$ determines also a homotopy $e_\alpha : \mathsf{Hot}(e_f, e_g)$. For $x : A$ and $u : B(x) \to C$, the component $(e_\alpha)_{x,u}$ of this homotopy is given by $e(x, u, \mathsf{int}(\alpha_u))$, where $\mathsf{int}(\alpha_u)$ is the path associated to the homotopy $(\lambda y : B(x))\alpha_{uy}$.

Let us now fix two P-algebra sections of E, $f = (f, \bar{f})$ and $g = (g, \bar{g})$.

Definition 4.7. A *P-algebra section homotopy* $(\alpha, \bar{\alpha}) : f \sim g$ is a homotopy $\alpha : \mathsf{Hot}(f, g)$ equipped with a homotopy $\bar{\alpha} : \mathsf{Hot}\big(e_\alpha \cdot \mathsf{ext}(\bar{f})\,, \mathsf{ext}(\bar{g}) \cdot (\alpha \circ \sup_C)\big)$.

The components of the homotopy $\bar{\alpha}$ that is part of a P-algebra section homotopy as above can be represented diagrammatically as fitting in the following diagram

$$\begin{array}{ccc} f(\sup_C(x,u)) & \xrightarrow{(\mathsf{ext}\,\bar{f})_{x,u}} & e(x,u,fu) \\ {\scriptstyle \alpha_{\sup_C(x,u)}}\downarrow & \Downarrow \bar{\alpha}_{x,u} & \downarrow{\scriptstyle e(x,u,\mathsf{int}(\alpha_u))} \\ g(\sup_C(x,u))) & \xrightarrow[(\mathsf{ext}\,\bar{g})_{x,u}]{} & e(x,u,gu)\,. \end{array}$$

Accordingly, we define the type of P-algebra homotopies of sections as follows:

$$\mathsf{AlgSecHot}((f,\bar{f}),\,(g,\bar{g})) =_{\mathrm{def}} (\Sigma\alpha : \mathsf{Hot}(f,g))\,\mathsf{Hot}\big(e_\alpha \cdot \mathsf{ext}(\bar{f})\,, \mathsf{ext}(\bar{g}) \cdot (\alpha \circ \sup_C)\big)\,.$$

Remarkably, in spite of the complexity of its definition, the notion of a P-algebra homotopy is equivalent to that of an identity proof between P-algebra sections, as the next lemma makes precise.

Lemma 4.8. *The canonical function*

$$\mathsf{ext}^{\mathsf{AlgSec}}_{(f,\bar{f}),(g,\bar{g})} : \mathsf{Id}\big((f,\bar{f}),\,(g,\bar{g})\big) \to \mathsf{AlgSecHot}\big((f,\bar{f}),\,(g,\bar{g})\big)$$

is an equivalence of types.

Proof. For $p : \mathsf{Id}(f,g)$ we have $p_!(\bar{f}) : \mathsf{Id}(g \circ \sup_C, e_g)$ and it can be shown by Id-elimination that there exists a path $q : \mathsf{Id}\big(e_p \cdot \bar{f}\,, p_!(\bar{f}) \cdot (p \circ \sup_C)\big)$, which can be represented with the diagram

$$\begin{array}{ccc} f \circ \sup_C & \xrightarrow{\bar{f}} & e_f \\ {\scriptstyle p \circ \sup_C}\downarrow & \Downarrow q & \downarrow{\scriptstyle e_p} \\ g \circ \sup_C & \xrightarrow[p_!(\bar{f})]{} & e_g\,. \end{array}$$

We then have

$$\begin{aligned} \mathsf{Id}\big((f,\bar{f}),(g,\bar{g})\big) &\simeq (\Sigma p : \mathsf{Id}(f,g))\ \mathsf{Id}(p_!(\bar{f})\,, \bar{g}) \\ &\simeq (\Sigma p : \mathsf{Id}(f,g))\ \mathsf{Id}\big(e_p \cdot \bar{f} \cdot (p \circ \sup_C)^{-1}\,, \bar{g}\big) \\ &\simeq (\Sigma p : \mathsf{Id}(f,g))\ \mathsf{Id}\big(e_p \cdot \bar{f}\,, \bar{g} \cdot (p \circ \sup_C)\big) \\ &\simeq (\Sigma p : \mathsf{Id}(f,g))\mathsf{Hot}\big(e_{\mathsf{ext}p} \cdot \mathsf{ext}(\bar{f})\,, \mathsf{ext}(\bar{g}) \cdot ((\mathsf{ext}\,p) \circ \sup_C)\big) \\ &\simeq (\Sigma \alpha : \mathsf{Hot}(f,g))\ \mathsf{Hot}\big(e_\alpha \cdot \mathsf{ext}(\bar{f})\,, \mathsf{ext}(\bar{g}) \cdot (\alpha \circ \sup_C)\big) \\ &= \mathsf{AlgSecHot}\big((f,\bar{f})\ (g,\bar{g})\big)\,. \end{aligned}$$

□

Note that Lemma 4.4, which we left without proof, follows as a special case of Lemma 4.8.

Algebra equivalences. We introduce the notion of equivalence between P-algebras. This will be useful in Section 5, where we will prove that assuming the univalence axiom, a form of univalence holds also for P-algebras.

Definition 4.9. We say that a P-algebra morphism $f : C \to D$ is a *P-algebra equivalence* if there exist P-algebra morphisms $g, h : D \to C$ which provide a left and a right P-inverse for f as a P-algebra morphism, i.e. for which there are paths of P-algebra morphisms

$$p : \mathsf{Id}_{\mathsf{Alg}(C,C)}(gf, 1_C)\,, \quad q : \mathsf{Id}_{\mathsf{Alg}(D,D)}(fh, 1_D)\,.$$

Given a P-algebra morphism $f : C \to D$, we define the type of proofs that f is an equivalence of P-algebras as follows:

$$\mathsf{isalgequiv}(f) =_{\mathrm{def}} (\Sigma g : \mathsf{Alg}(D,C))\mathsf{Id}_{\mathsf{Alg}(C,C)}(gf, 1_C) \times (\Sigma h : \mathsf{Alg}(D,C))\mathsf{Id}_{\mathsf{Alg}(D,D)}(fh, 1_D)\,.$$

We then define the type of P-algebra equivalences between C and D as

$$\mathsf{AlgEquiv}(C,D) =_{\mathrm{def}} (\Sigma f : \mathsf{Alg}(C,D))\,\mathsf{isalgequiv}(f)\,.$$

Lemma 4.10. *The underlying function of a P-algebra equivalence is an equivalence, i.e. for every P-algebra morphism $(f, \bar{f}) : C \to D$ there is a function*

$$\pi_f : \mathsf{isalgequiv}(f, \bar{f}) \to \mathsf{isequiv}(f)\,.$$

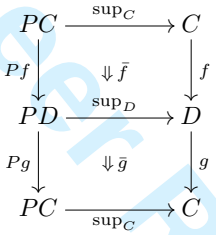
Proof. Let $(f, \bar{f}) : C \to D$ be a P-algebra morphism. Unfolding the definition, we have

$$\mathsf{isalgequiv}(f, \bar{f}) =_{\mathrm{def}} (\Sigma g : D \to C)\big(\Sigma \bar{g} : \mathsf{isalghom}(g)\big)\, G(g, \bar{g}) \times (\Sigma h : D \to C)\big(\Sigma \bar{h} : \mathsf{isalghom}(h)\big)\, H(h, \bar{h})\,,$$

where we used the notation introduced in (4.3) and

$$G(g,\bar{g}) =_{\mathrm{def}} \mathsf{Id}\big((gf, \overline{gf}), (1_C, \bar{1}_C)\big)\,, \quad H(h,\bar{h}) =_{\mathrm{def}} \mathsf{Id}\big((fh, \overline{fh}), (1_D, \bar{1}_D)\big)\,.$$

The types $G(g,\bar{g})$ and $H(h,\bar{h})$ can be thought of as the types of proofs that $(g,\bar{g})$ and $(h,\bar{h})$ are a left and right inverse for $(f,\bar{f})$ as P-algebra morphisms, respectively. For the right inverse, this amounts to requiring that the pasting diagram



is propositionally equal to the diagram for the identity P-algebra morphism on C in (4.4). By the characterization of paths in Σ-types, we have

$$G(g,\bar{g}) \simeq (\Sigma p : \mathsf{Id}(gf, 1_C))\,\mathsf{Id}\big(\overline{gf},\, p^*(\bar{1}_C)\big)\,, \quad H(h,\bar{h}) \simeq (\Sigma q : \mathsf{Id}(fh, 1_C))\,\mathsf{Id}\big(\overline{fh},\, q^*(\bar{1}_C)\big)\,.$$

Thus, rearranging the Σ-types in the definition, we have

$$\mathsf{isalgequiv}(f, \bar{f}) \simeq (\Sigma g : D \to C)(\Sigma p : \mathsf{Id}(gf, 1_C))\, G'(g,p) \times (\Sigma h : D \to C)(\Sigma q : \mathsf{Id}(fh, 1_C))\, H'(h,q)\,, \quad (4.6)$$

where

$$G'(g,p) =_{\mathrm{def}} (\Sigma \bar{g} : \mathsf{isalghom}(g))\ \mathsf{Id}(\overline{gf}, p^*(\bar{1}_C))\,, \quad (4.7)$$

$$H'(h,q) =_{\mathrm{def}} (\Sigma \bar{h} : \mathsf{isalghom}(h))\ \mathsf{Id}(\overline{fh}, q^*(\bar{1}_D)))\,. \quad (4.8)$$

The canonical elements of $G(g,p)$ are pairs $(\bar{g}, \bar{p})$ consisting of a path $\bar{g}$ making g into a P-algebra morphism and a path $\bar{p}$ making $p : \mathsf{Id}(gf, 1_C)$ into a propositional equality between the P-algebra morphisms $(gf, \overline{gf})$ and $(1_C, \bar{1}_C)$. It is now clear that we can obtain the required function π_f by composing the equivalence in (4.6) with the evident projections and the equivalence in (1.9). □

Proposition 4.11 below can be understood informally as saying that for a P-algebra morphism f, there is an essentially unique way of turning an inverse of f as a function into an inverse of f as a P-algebra morphism.

Proposition 4.11. *A P-algebra morphism $(f, \bar{f}) : C \to D$ is an equivalence of P-algebras if and only if its underlying function $f : C \to D$ is an equivalence of types, i.e. the function*

$$\pi_f : \mathsf{isalgequiv}(f, \bar{f}) \to \mathsf{isequiv}(f)$$

is an equivalence.

Proof. Let $(f, \bar{f}) : (C, \mathrm{sup}_C) \to (D, \mathrm{sup}_D)$ be a P-algebra morphism. We will show that all the homotopy fibers of the function π_f are contractible. So, let us consider a canonical element of the codomain of π_f, given by a 4-tuple $(g, h, p, q) : \mathsf{isequiv}(f)$ consisting of functions $g : D \to C$ and $h : D \to C$ and paths $p : \mathsf{Id}(gf, 1_C)$, $q : \mathsf{Id}(fh, 1_D)$, exhibiting g and h as a right and a left inverse of f (as a function, not as a P-algebra morphism), respectively.

The homotopy fiber of π_f over this element can be thought of as the type consisting of all the data that is missing from having a left and a right inverse of f as a P-algebra morphism. In particular, we have

$$\mathsf{hfiber}(\pi_f, (g, h, p, q)) \simeq G'(g, p) \times H'(h, q)\,,$$

where $G'(g, p)$ and $H'(h, q)$ are defined as in (4.7) and (4.8), respectively. Therefore, it suffices to prove that $G'(g, p)$ and $H'(h, q)$ are contractible. The proofs that $G'(g, p)$ and $H'(h, q)$ are contractible are essentially identical, so we consider only $G'(h, p)$.

First of all, recall that the path $\overline{gf} : \mathsf{isalghom}(gf)$ is given by $\overline{gf} =_{\mathrm{def}} (g \circ \bar{f}) \cdot (\bar{g} \circ Pf)$, where we suppressed the path relative to the pseudo-functoriality of P, as in (4.1), for convenience. By Id-elimination on p, the path $p^*(\bar{1}_C) : \mathsf{isalghom}(gf)$ is propositionally equal to the composite path

$$(gf) \circ \mathrm{sup}_C \xrightarrow{p \circ \mathrm{sup}_C} 1_C \circ \mathrm{sup}_C \xrightarrow{\bar{1}_C} \mathrm{sup}_C \circ P(1_C) \xrightarrow{\mathrm{sup}_C \circ P(p^{-1})} \mathrm{sup}_C \circ P(gf)\,.$$

Hence, we have

$$\begin{aligned} G(g, p) &\simeq (\Sigma \bar{g} : \mathsf{AlgHom}(g))\ \mathsf{Id}\big((g \circ \bar{f}) \cdot (\bar{g} \circ Pf)\,, (\mathrm{sup}_C \circ P(p^{-1})) \cdot \bar{1}_C \cdot (p \circ \mathrm{sup}_C)\big) \\ &\simeq (\Sigma \bar{g} : \mathsf{AlgHom}(g))\ \mathsf{Id}\big(\bar{g} \circ Pf\,, (g \circ \bar{f})^{-1} \cdot (\mathrm{sup}_C \circ P(p^{-1})) \cdot \bar{1}_C \cdot (p \circ \mathrm{sup}_C)\big)\,. \end{aligned}$$

Now, since $f : C \to D$ is an equivalence, $Pf : PC \to PD$ is also an equivalence and hence so is the function mapping a path $r : \mathsf{Id}_{PD \to C}(s, t)$ to the composite $r \circ Pf : \mathsf{Id}_{PC \to C}(s \circ Pf, t \circ Pf)$. Thus, by part (ii) of Lemma 2.11, $G'(g, p)$ is contractible, as required. □

Corollary 4.12. *For every P-algebra morphism $(f, \bar{f})$, the type* $\mathsf{isalgequiv}(f, \bar{f})$ *is a mere proposition.* □

5. Homotopy-initial algebras

Inductive algebras. Given a P-algebra $C = (C, \mathrm{sup}_C)$ and a type D, an equivalence of types $f : C \to D$ makes D into a P-algebra with structure map $\mathrm{sup}_D : PD \to D$ given by the composite

$$PD \xrightarrow{P(f^{-1})} PC \xrightarrow{\mathrm{sup}_C} C \xrightarrow{f} D\,,$$

where $f^{-1} : D \to C$ is a quasi-inverse of $f : C \to D$. In particular, for $W = (\mathrm{W}x : A)B(x)$, if we have an equivalence $f : W \to D$, then the induced P-algebra structure $\mathrm{sup}_D : PD \to D$ defined as above is such that D also satisfies a form of the elimination rule for W-types. We shall see that D satisfies the other rules as well, but with a weakened computation rule.

Definition 5.1. We say that a P-algebra C is *inductive* if every fibered P-algebra over it has a P-algebra section, i.e. the type

$$\mathsf{isind}(C) =_{\mathrm{def}} (\Pi E : \mathsf{FibAlg}(C))\ \mathsf{AlgSec}(C, E)$$

is inhabited.

In complete analogy with the case of bipointed types, for a P-algebra C, the type $\mathsf{isind}(C)$ is a mere proposition. We also have the following analogue of Proposition 3.7.

Proposition 5.2. *Homotopy-initial P-algebras are unique up to a contractible type of algebra equivalences, i.e. the type*

$$(\Pi C : \mathsf{Alg})(\Pi D : \mathsf{Alg})\big(\mathsf{ishinit}(C) \times \mathsf{ishinit}(D) \to \mathsf{iscontr}(\mathsf{AlgEquiv}(C, D))\big)\,.$$

is inhabited.

Proof. Let C and D be P-algebras. The type $\mathsf{Alg}(C, D)$ is contractible by homotopy-initiality of C. Since the dependent sum of a family of mere propositions over a mere proposition is again a mere proposition, it suffices to prove $\mathsf{iscontr}(\mathsf{isalgequiv}(f))$ for any P-algebra morphism f. This type is a mere proposition, as remarked earlier; thus it suffices to show it is inhabited. Since D is homotopy-initial, there exists a P-algebra morphism $g : D \to C$. Again by homotopy-initiality of C and D, we have $\mathsf{Id}(g \circ f, 1_C)$ and $\mathsf{Id}(f \circ g, 1_D)$, which gives us the desired P-algebra equivalence between C and D. □

The next proposition characterizes inductive P-algebras by means of deduction rules, where we display premisses in multiple lines for lack of space.

Proposition 5.3. *Let $C = (C, \mathrm{sup}_C)$ be a P-algebra. Then C is inductive if and only if we can derive rules of the form*

(i) the elimination rule,

$$\frac{\begin{array}{rcl} z : C & \vdash & E(z) : \mathsf{U} \\ x : A\,, u : B(x) \to C,\, v : (\Pi y : B(x))E(uy) & \vdash & e(x, u, v) : E(\mathrm{sup}_C(x, u)) \end{array}}{z : C \vdash \mathsf{elim}(z, e) : E(z)}$$

(ii) the computation rule,

$$\frac{\begin{array}{rcl} z : C & \vdash & E(z) : \mathsf{U} \\ x : A,\, u : B(x) \to C,\, v : (\Pi y : B(x))E(uy) & \vdash & e(x, u, v) : E(\mathrm{sup}_C(x, u)) \end{array}}{x : A,\, u : B(x) \to C \vdash \mathsf{comp}(x, u, e) : \mathsf{Id}\big(\mathsf{elim}(\mathrm{sup}_C(x, u), e), e(x, u, (\lambda y : B(x))\mathsf{elim}(uy, e))\big)\,.}$$

Proof. The rules are simply an unfolding of the definition of an inductive algebra. □

Below, when working with an inductive P-algebra, we will always assume to have constants elim and comp as in Proposition 5.3. We now show the essential uniqueness of algebra sections of inductive fibered algebras.

Proposition 5.4. *Let $C = (C, \mathrm{sup}_C)$ be a P-algebra. If C is inductive, then we can derive rules of the following form:*

(i) the η-rule,

$$\frac{\begin{array}{rcl} z : C & \vdash & E(z) : \mathsf{U} \\ x : A,\, u : B(x) \to C,\, e : (\Pi y : B(x))E(uy) & \vdash & e(x, u, v) : E(\mathrm{sup}_C(x, u)) \\ z : C & \vdash & f(z) : E(z) \\ x : A\,, u : B(x) \to C & \vdash & \phi_{x,u} : \mathsf{Id}\big(f(\mathrm{sup}_C(x, u)), e\big(x, u, fu\big)\big) \end{array}}{z : C \vdash \eta_z : \mathsf{Id}(f(z), \mathsf{elim}(z, e))}$$

(ii) the coherence rule,

$$\frac{\begin{array}{rcl} z:C & \vdash & E(z):\mathsf{U} \\ x:A,\, u:B(x)\to C,\, v:(\Pi y:B(x))E(uy) & \vdash & e(x,u,v):E(\sup_C(x,u)) \\ z:C & \vdash & f(z):E(z) \\ x:A\,,u:B(x)\to C & \vdash & \phi_{x,u}:\mathsf{Id}\big(f(\sup_C(x,u)),e\big(x,u,fu\big)\big) \end{array}}{x:A,u:B(x)\to C\vdash \bar{\eta}_{x,u}:\mathsf{Id}\big(\eta_{\sup_C(x,u)}\cdot \mathsf{comp}(x,u,e),\ \phi_{x,u}\cdot e(x,u,\mathsf{int}(\eta_u))\big)}$$

Before proving the proposition, observe that the paths $\bar{\eta}_{x,u}$ in the conclusion of the coherence rule can be seen as fitting in the diagram

$$\begin{array}{ccc} f(\sup_C(x,u)) & \xrightarrow{\quad\eta_{\sup_C(x,u)}\quad} & \mathsf{elim}(\sup_C(x,u),e)) \\ {\scriptstyle\phi_{x,u}}\downarrow & \Downarrow\bar{\eta}_{x,u} & \downarrow{\scriptstyle\mathsf{comp}(x,u,e)} \\ e(x,u,fu) & \xrightarrow[e(x,u,\mathsf{int}(\eta_u))]{} & e(x,u,\mathsf{elim}(x,u,(\lambda y:B(x))\,\mathsf{elim}(uy,e))) \end{array}$$

Proof of Proposition 5.4. For $z:C$, let us define $T(z)=_{\mathrm{def}}\mathsf{Id}\big(f(z),\mathsf{elim}(z,e)\big)$. With this notation, proving the η-rule amounts to defining $\eta_z:T(z)$, for $z:C$. In order to do so, we apply the elimination rule for C. We need to show that, for $x:A$, $u:B(x)\to C$ and $v:(\Pi y:B(x))\,T(uy)$, there is

$$t(x,u,v):T(\sup_C(x,u))\,.$$

Note that v is a homotopy between fu and $(\lambda y:B(x))\,\mathsf{elim}(uy,e)$. Hence, we have a corresponding path $\mathsf{int}(v)$. We can construct the required path as follows:

$$\begin{array}{lll} f(\sup_C(x,u)) \cong e\big(x,u,fu\big) & & \text{by } \phi_{x,u} \\ \qquad\cong e\big(x,u,(\lambda y:B(x))\,\mathsf{elim}(uy,e)\big) & & \text{by } \mathsf{int}(v) \\ \qquad\cong \mathsf{elim}(\sup_C(x,u),e) & & \text{by } \mathsf{comp}(x,u,e)^{-1}. \end{array}$$

For $z:C$, we can then define

$$\eta_z=_{\mathrm{def}}\mathsf{elim}(z,t)\,.$$

For $x:A$ and $u:B(x)\to C$, the computation rule of Proposition 5.3 then gives us

$$\eta_{\sup_C(x,u)}\cong\phi_{x,u}\cdot e(x,y,\mathsf{int}(\eta_u))\cdot\mathsf{comp}(x,u,e)^{-1}\,.$$

The path required to prove the coherence rule is then obtained using the groupoid laws. □

Corollary 5.5. *For every P-algebra C, the type* $\mathsf{isind}(C)$ *is a mere proposition.*

Proof. Analogous to that of Corollary 3.4. □

Homotopy-initial algebras. Exactly as in the case of bipointed types, the hypothesis that a P-algebra C is inductive allows us to show that for any P-algebra D, there is a P-algebra morphism $f:C\to D$ which is unique up to a P-algebra path, itself is unique up to a higher path, which in turn is unique up to a yet higher path, and so on. As before, we shall characterize this kind of universal property using the notion of a homotopy-initial P-algebra, which we define next.

Definition 5.6. Let $C=(C,\sup_C)$ be a P-algebra. We say that C is *homotopy-initial* if for any P-algebra $D=(D,\sup_D)$, the type $\mathsf{Alg}(C,D)$ of P-algebra morphisms from C to D is contractible, i.e. the following type is inhabilited

$$\mathsf{ishinit}(C)=_{\mathrm{def}}(\Pi D:\mathsf{Alg})\,\mathsf{iscontr}\big(\mathsf{Alg}(C,D)\big)\,.$$

We stress again that homotopy-initiality is a purely type-theoretic notion. Also note that, exactly as for homotopy-initiality of bipointed types, for a P-algebra C, the type $\mathsf{ishinit}(C)$ is a mere proposition. We have the following type-theoretic analogue of Lambek's lemma, which will be used in the proof of Proposition 5.14 below.

Lemma 5.7. *Let $C = (C, \mathrm{sup}_C)$ be a P-algebra. If C is homotopy-initial, then the structure map $\mathrm{sup}_C : PC \to C$ is an equivalence.*

Proof. This is a straightforward translation of the standand category-theoretic proof, but we provide some details to illustrate where the contractibility condition in the definition of a homotopy-initial algebra is used. For brevity, let us write $s : PC \to C$ for the structure map of C.

We wish to construct a quasi-inverse to $s : PC \to C$. In order to do so, we use the homotopy-initiality of C. First of all, observe that PC can be made into a P-algebra by considering the structure map $Ps : PPC \to PC$. Thus, by the contractibility of the type $\mathsf{Alg}(C, PC)$, there exists a P-algebra morphism $(t, \bar{t}) : C \to PC$. We represent it as the diagram

$$\begin{array}{ccc} PC & \xrightarrow{\quad s \quad} & C \\ {\scriptstyle Pt}\downarrow & \Downarrow \bar{t} & \downarrow{\scriptstyle t} \\ PPC & \xrightarrow[\quad Ps \quad]{} & PC \end{array}$$

Now, the composite $st : C \to C$ and the identity $1_C : C \to C$ are both P-algebra morphisms and so, by the contractibility of $\mathsf{Alg}(C, C)$, there has to be a path $p : \mathsf{Id}(s \circ t, 1_C)$. Using this fact, we can also show that there is a path $q : \mathsf{Id}(ts, 1_{PC})$. Indeed, we have

$$t \circ s \cong Ps \circ Pt \cong P(s \circ t) \cong P(1_C) \cong 1_{PC}\,,$$

where the first path is given by $\bar{t}$, the second by the pseudo-functoriality of P, as in (4.1), the third is the path p constructed above, and the fourth one is given again by the pseudo-functoriality of P, as in (4.1). □

Proposition 5.8. *A P-algebra $C = (C, \mathrm{sup}_C)$ is homotopy-initial if and only if we can derive rules of the following form:*

(i) the recursion rule,

$$\frac{D : \mathsf{U} \qquad x : A\,, u : B(x) \to D \vdash \mathrm{sup}_D(x, u) : D}{z : C \vdash \mathsf{rec}(z, \mathrm{sup}_D) : D}$$

(ii) the β-rule,

$$\frac{D : \mathsf{U} \qquad x : A\,, u : B(x) \to D \vdash \mathrm{sup}_D(x, u) : D}{x : A,\ u : B(x) \to D \vdash \beta(x, u, \mathrm{sup}_D) : \mathsf{Id}\big(\mathsf{rec}(\mathrm{sup}_C(x, u), \mathrm{sup}_D)\,, \mathrm{sup}_D\big(x, (\lambda y : B(x))\, \mathsf{rec}(uy, \mathrm{sup}_D)\big)\big)}$$

(iii) the η-rule,

$$\frac{\begin{array}{rcl} & & D : \mathsf{U} \\ x : A,\ u : B(x) \to D & \vdash & \mathrm{sup}_D(x, u) : D \\ z : C & \vdash & f(z) : D \\ x : A, u : B(x) \to D & \vdash & \phi_{x,u} : \mathsf{Id}(f(\mathrm{sup}_C(x, u)), \mathrm{sup}_D(x, fu)) \end{array}}{z : A \vdash \eta_z : \mathsf{Id}(f(z), \mathsf{rec}(z, \mathrm{sup}_D))}$$

(iv) the (β,η)-coherence rule,

$$\frac{\begin{array}{rcl} & & D:\mathsf{U} \\ x:A,\, u:B(x)\to D & \vdash & \sup_D(x,u):D \\ z:C & \vdash & f(z):D \\ x:A,u:B(x)\to D & \vdash & \phi_{x,u}:\mathsf{Id}(f(\sup_C(x,u)),\sup_D(x,f\circ u)) \end{array}}{x:A,u:B(x)\to C\vdash \bar\eta_{x,u}:\mathsf{Id}(\beta(x,u,\sup_D)\cdot\eta_{\sup_C(x,u)}\,,\sup_D(x,\mathsf{int}(\eta_u))\cdot\phi_{x,u})}$$

Proof. The rules can be read as follows. The recursion rule says that, given any type D together with the function $\sup_D : PD \to D$, i.e. any P-algebra, there is a function $r : C \to D$ defined by letting, for $z : C$, $r(z) = \mathsf{rec}(z, \sup_D)$. The β-rule implies that we have a homotopy $\beta : \mathsf{Hot}(r \circ \sup_C\,, \sup_D \circ Pr)$ and so, by function extensionality, we get a path $\bar r$ fitting in the diagram

$$\begin{array}{ccc} PC & \xrightarrow{\sup_C} & C \\ {\scriptstyle r}\downarrow & \Downarrow \bar r & \downarrow{\scriptstyle r} \\ PD & \xrightarrow[\sup_D]{} & D\,. \end{array}$$

We therefore obtain a P-algebra morphism $(r,\bar r) : C \to D$. The η-rule says that if $f : C \to D$ is a P-algebra morphism, then there is a homotopy $\eta : \mathsf{Hot}(f, r)$. And the (β,η)-compatibility rule says that η is in fact a P-algebra homotopy. Using again Proposition 4.8, this shows that there is a path from $(r,\bar r)$ to $(f,\bar f)$, thus proving the contractibility of $\mathsf{Alg}(C,D)$. □

Remark 5.9. As for bipointed types, the special case of the rules in Proposition 5.8 obtained by considering $C = D$ and $f = 1_C$ provides some explanation for the terminology used to denote them. By the recursion rule, we obtain a function $r : C \to C$ defined by $r = (\lambda z : C)\mathsf{rec}(z, \sup_C)$. The β-rule gives a homotopy with components $\beta_{x,u} : \mathsf{Id}(r(\sup_C(x,u)), \sup_C(x, ru))$, the η-rule gives a homotopy with components $\eta_z : \mathsf{Id}(z, r(z))$ and, finally, the (β,η)-coherence rule, gives us a homotopy with components fitting in the diagram

$$\begin{array}{ccc} \sup_C(x,u) & \xrightarrow{\eta_{\sup_C(x,u)}} & r(\sup_C(x,u)) \\ & \overset{\bar\eta_{x,u}}{\Rightarrow} & \downarrow{\scriptstyle \beta_{x,u}} \\ & {\scriptstyle \sup_C(x,\mathsf{int}(\eta_u))}\searrow & \sup_C(x,ru)\,. \end{array}$$

We can now state and prove our main result.

Theorem 5.10. *A P-algebra is inductive if and only if it is homotopy-initial, i.e. the type*

$$(\Pi C : \mathsf{Alg})\big(\mathsf{isind}(C) \leftrightarrow \mathsf{ishinit}(C)\big)$$

is inhabited.

Proof. Let $C = (C, \sup_C)$ be an inductive P-algebra. We wish to show that it is homotopy initial. For this, it suffices to observe that the rules in Proposition 5.8 characterizing homotopy-initial algebras are a special case of those given in Proposition 5.3 and Proposition 5.4.

For the converse, we proceed as in the proof of Theorem 3.10. Let $E = (E, e)$ be a fibered algebra over C. We need to show that there exists a P-algebra section $(s,\bar s)$, where $s : (\Pi x : C)E(x)$ and

$$\bar s : (\Pi x : A)(\Pi u : B(x) \to C)\mathsf{Id}\big(s(\sup_C(x,u)), e(x,u,su)\big)$$

We consider the P-algebra $(E', \sup_{E'})$ associated to E. Recall that $E' =_{\text{def}} (\Sigma z : C)E(z)$ and $\sup_{E'} : PE' \to E'$ is defined so that, for $x : A$ and $u : B(x) \to E'$, we have

$$\sup_{E'}(x, u) = \left(\sup_C(x, \pi_1 u)\,, e(x, \pi_1 u, \pi_2 u)\right).$$

In this way, the first projection $\pi_1 : E' \to C$ is an algebra morphism, represented by the diagram

$$\begin{array}{ccc} PE' & \xrightarrow{\sup_{E'}} & E' \\ {\scriptstyle P\pi_1}\downarrow & \Downarrow \overline{\pi}_1 & \downarrow{\scriptstyle \pi_1} \\ PC & \xrightarrow[\sup_C]{} & C\,. \end{array}$$

By the homotopy-initiality of C, there exists an algebra morphism $(f, \bar{f}) : (C, \sup_C) \to (E', \sup_{E'})$, which we represent with the diagram

$$\begin{array}{ccc} PC & \xrightarrow{\sup_C} & C \\ {\scriptstyle Pf}\downarrow & \Downarrow \bar{f} & \downarrow{\scriptstyle f} \\ PE' & \xrightarrow[\sup_{E'}]{} & E'\,. \end{array}$$

Let $\phi =_{\text{def}} (\mathsf{ext}\,\bar{f})$ be the homotopy associated to the path $\bar{f}$. We write $f_1 : C \to C$ for the composite $\pi_1 f : C \to C$, which is a P-algebra morphism. The path

$$\begin{array}{ccc} PC & \xrightarrow{\sup_C} & C \\ {\scriptstyle Pf_1}\downarrow & \Downarrow \bar{f}_1 & \downarrow{\scriptstyle f_1} \\ PC & \xrightarrow[\sup_C]{} & C \end{array}$$

is given by the pasting diagram

$$\begin{array}{ccc} PC & \xrightarrow{\sup_C} & C \\ {\scriptstyle Pf}\downarrow & \Downarrow \bar{f} & \downarrow{\scriptstyle f} \\ PE & \xrightarrow[\sup_{E'}]{} & E' \\ {\scriptstyle P\pi_1}\downarrow & \Downarrow \bar{\pi}_1 & \downarrow{\scriptstyle \pi_1} \\ PC & \xrightarrow[\sup_C]{} & C\,. \end{array}$$

Let $\phi_1 =_{\text{def}} (\mathsf{ext}\,\bar{f}_1)$ be the homotopy associated to the path $\bar{f}_1$. Unfolding the definitions, we have that, for $x : A, u : B(x) \to C$,

$$(\phi_1)_{x,u} \cong \pi_1\, \mathsf{ext}^{\Sigma}\, \phi_{x,u}\,. \tag{5.1}$$

Furthermore, let us define $f_2 : (\Pi z : C)E(f_1 z)$ by setting

$$f_2 =_{\text{def}} (\lambda z : C)\, \pi_2 f z\,.$$

In order to define the required section, observe that, by the homotopy-initiality of C and Lemma 4.4, there exists a P-algebra homotopy

$$(\alpha, \bar{\alpha}) : \mathsf{AlgHot}(f_1, 1_C)\,,$$

where $\alpha : \mathsf{Hot}(f_1, 1_C)$ and, for $x : A$, $u : B(x) \to C$, the path $\bar{\alpha}_{x,u}$ fits into the diagram

$$\begin{array}{ccc} f_1 \mathrm{sup}_C(x,u) & \xrightarrow{(\phi_1)_{x,u}} & \mathrm{sup}_C(x, f_1 u) \\ {\scriptstyle \alpha_{\mathrm{sup}_C(x,u)}}\downarrow & \Downarrow \bar{\alpha}_{x,u} & \downarrow{\scriptstyle \mathrm{sup}_C(x, \mathsf{int}(\alpha_u))} \\ \mathrm{sup}_C(x,u) & \xrightarrow[\bar{1}_{x,u}]{} & \mathrm{sup}_C(x,u)\,. \end{array} \tag{5.2}$$

Here, $\mathsf{int}(\alpha_u) : \mathsf{Id}(f_1 u\,, u)$ is the path associated to the homotopy $(\lambda y : B(x))\alpha_{uy} : \mathsf{Hot}(f_1 u, u)$. We define the required section $s : (\Pi z : C)E(z)$ so that, for $z : C$ we have

$$sz =_{\mathrm{def}} (\alpha_z)_!\big(f_2 z\big)\,,$$

where $(\alpha_z)_! : E(f_1 z) \to E(z)$ is a transport map associated to the path $\alpha_z : \mathsf{Id}(f_1 z, z)$. It now suffices to define, for each $x : A$ and $u : B(x) \to C$, a path

$$\bar{s}(x,u) : \mathsf{Id}\big(s(\mathrm{sup}_C(x,u))\,, e_s(x,u)\big)\,,$$

where e_s is defined using the formula in (4.5). Unfolding the definitions, our goal is to show that

$$(\alpha_{\mathrm{sup}_C(x,u)})_!(f_2\,\mathrm{sup}_C(x,u)) \cong e(x,u,(\lambda y : B(x))(\alpha_{uy})_!(f_2 uy))\,. \tag{5.3}$$

Our goal will follow once we show the following:

Claim 1. $\alpha_{\mathrm{sup}_C(x,u)} \cong \mathrm{sup}_C(x, \mathsf{int}(\alpha_u)) \cdot (\phi_1)_{x,u}$

Claim 2. $((\phi_1)_{x,u})_!(f_2\,\mathrm{sup}_C(x,u)) \cong e(x, f_1 u, f_2 u)\,.$

Claim 3. $(\mathrm{sup}_C(x, \mathsf{int}(\alpha_u)))_!\, e(x, f_1 u, f_2 u) \cong e\big(x, u, (\lambda y : B(x))(\alpha_{uy})_!(f_2 uy)\big)\,.$

Inded, the required propositional equality in (5.3) can then be obtained as follows:

$$\begin{aligned} (\alpha_{\mathrm{sup}_C(x,u)})_!(f_2\,\mathrm{sup}_C(x,u)) &\cong (\mathrm{sup}_C(x, \mathsf{int}(\alpha_u)))_!\ ((\phi_1)_{x,u})_!(f_2\,\mathrm{sup}_C(x,u)) && \text{(by Claim 1)} \\ &\cong (\mathrm{sup}_C(x, \mathsf{int}(\alpha_u)))_!\ e(x, f_1 u, f_2 u) && \text{(by Claim 2)} \\ &\cong e(x,u,(\lambda y : B(x))(\alpha_{uy})_!(f_2 uy)) && \text{(by Claim 3).} \end{aligned}$$

We conclude by proving the auxiliary claims stated above.

Proof of Claim 1. This follows by the path in the diagram in (5.2).

Proof of Claim 2. Recall that the homotopy ϕ has components

$$\phi_{x,u} : \mathsf{Id}\big(f\,\mathrm{sup}_C(x,u)\,, \mathrm{sup}_{E'}(x, fu)\big)$$

Thus, by the characterization of paths in Σ-types, we have

$$p : \mathsf{Id}(f_1 \mathrm{sup}_C(x,u), \mathrm{sup}_C(x, f_1 u))\,, \quad q : \mathsf{Id}\big(p_!(f_2\,\mathrm{sup}_C(x,u))\,, e(x, f_1 u, f_2 u)\big)\,,$$

where $p =_{\mathrm{def}} \pi_1\,\mathsf{ext}^{\Sigma}\phi_{x,u}$ and $q =_{\mathrm{def}} \pi_2\,\mathsf{ext}^{\Sigma}\,\phi_{x,u}$. The claim now follows by (5.1).

Proof of Claim 3. Observe that for all $a : A$, $p : \mathsf{Id}_{B(a)\to C}(t_1, t_2)$ and $v : (\Pi y : B(a))E(t_1 y)$, we have

$$(\mathrm{sup}_C(a,p))_!\, e(a, t_1, v) \cong e\big(a, t_2, (\lambda y : B(a))((\mathsf{ext}\, p)_y)_!\, vy\big)\,.$$

by Id-elimination. If we apply this to $x : A$, $\mathsf{int}(\alpha_u) : \mathsf{Id}(f_1 u, u)$, and $f_2 u : (\Pi y : B(x))E(f_1 uy)$, we get

$$\big(\mathrm{sup}_C(x, \mathsf{int}(\alpha_u))\big)_!\ e(x, f_1 u, f_2 u) \cong e\big(x, u, (\lambda y : B(x))\big((\alpha_{uy})_!\ f_2 uy\big)\big)\,,$$

as required. □

Corollary 5.11. *For every P-algebra C, there is an equivalence* $\mathsf{isind}(C) \simeq \mathsf{ishinit}(C)$.

Proof. Theorem 5.10 gives us a logical equivalence, but both types are mere propositions. □

Below, when we assume the rules for W-types (as in Table 7), we always write W for $(\mathrm{W}x : A)B(x)$.

Corollary 5.12. *Assuming the rules for W-types, for a P-algebra C the following conditions are equivalent:*

(i) C is inductive,
(ii) C is homotopy initial,
(iii) C is equivalent to W as a P-algebra.

In particular, the type W is a homotopy-initial P-algebra. □

Corollary 5.12 provides the analogue in our setting of the characterization of W-types as a strict initial algebra in extensional type theory. It makes precise the informal idea that, in intensional type theory, W-types are a kind of initial algebra in the weak $(\infty, 1)$-category of types, functions, paths and higher paths.

Lemma 5.13. *Assuming the rules for W-types, for all $a_1, a_2 : A$, $t_1 : B(x_1) \to W$, $t_2 : B(x_2) \to W$, there is an equivalence of types*

$$\mathsf{Id}_W(\sup_W(a_1, t_1), \sup_W(a_2, t_2)) \simeq \mathsf{Id}_{PW}\big((a_1, t_1), (a_2, t_2)\big)\,.$$

Proof. By Lemma 5.7 and Corollary 5.12, $\sup_W : PW \to W$ is an equivalence. □

We remark that W-types preserve homotopy levels, as already shown by N. A. Danielsson.[6]

Proposition 5.14 (Danielsson)**.** *Assuming the rules for W-types, if A has h-level $n+1$, then so does the W-type $(\mathrm{W}x : A)B(x)$.*

Proof. We need to show that for all $w, w' : W$ the type $\mathsf{Id}_W(w, w')$ has h-level n. We do so applying the elimination rule for W-types on $w : W$. So, let $x : A, u : B(x) \to W$ and assume the induction hypothesis

$(*)$ for every $y : B(x)$, for every $w' : W$, the type $\mathsf{Id}_W(uy, w')$ has h-level n,

and show that for every $w' : W$ the type $\mathsf{Id}(\sup_W(x, u), w')$ has h-level n. We apply again the elimination rule for W-types. So, let $x' : A$, $u' : B(x') \to W$ and assume the induction hypothesis (which we do not spell out since we will not need it) and show that $\mathsf{Id}(\sup_W(x, u), \sup_W(x', u'))$ has h-level n. We have

$$\begin{aligned}
\mathsf{Id}_W(\sup_W(x, u), \sup_W(x', u')) &\simeq \mathsf{Id}_{PW}((x, u), (x', u')) \\
&\simeq (\Sigma p : \mathsf{Id}_A(x, x'))\, \mathsf{Id}_{B(x)\to W}(u, p^*(u')) \\
&\simeq (\Sigma p : \mathsf{Id}_A(x, x'))\, \mathsf{Id}\big(u, (\lambda y : B(x))\, u'(p_! \, y)\big) \\
&\simeq (\Sigma p : \mathsf{Id}_A(x, x'))(\Pi y : B(x))\, \mathsf{Id}_W\big(uy, u'(p_! \, y)\big)\,.
\end{aligned}$$

Here, the first equivalence follows by Lemma 5.13 and the other equivalences follow by standard properties of the transport functions. Since A has h-level $n+1$ by assumption, we have that $\mathsf{Id}_A(x, x')$ has h-level n. Also, for any $p : \mathsf{Id}_A(x, x')$ and $u : B(x) \to W$, the type $\mathsf{Id}_W(uy, u'(p_! \, y))$ has h-level n by the induction hypothesis in $(*)$. The claim follows by recalling that the h-levels are closed under arbitrary dependent products and under dependent sums over types of the same h-level. □

We note that the h-level of $(\mathrm{W}x : A)B(x)$ does not depend on that of $B(x)$. Furthermore, assuming that we have a unit type 1, the lemma is no longer true if $n+1$ is replaced by n, as

[6] Post on the Homotopy Type Theory blog, 2012.

the following example illustrates: if $A =_{\mathrm{def}} 1$ and $B(x) =_{\mathrm{def}} 1$, then $(\mathrm{W}x : A)B(x) \simeq 0$, which is not contractible.

Univalence for algebras. We conclude the paper with some applications of the univalence axiom. The first is that, just as for bipointed types, a form of univalence holds also for P-algebras, as the next theorem makes precise.

Theorem 5.15. *Assuming the univalence axiom, the canonical function*

$$\mathsf{ext}^{\mathsf{Alg}}_{C,D} : \mathsf{Id}\big(C, D\big) \to \mathsf{AlgEquiv}(C, D)$$

is an equivalence for every pair of P-algebras C and D.

Proof. Let $C = (C, \mathrm{sup}_C)$ and $D = (D, \mathrm{sup}_D)$ be P-algebras. By the characterization of paths in Σ-types, $\mathsf{Id}\big((C, \mathrm{sup}_C), (D, \mathrm{sup}_D)\big)$ can be expressed as the type

$$(\Sigma p : \mathsf{Id}(C, D))\, \mathsf{Id}\big(\mathrm{sup}_C, p^*(\mathrm{sup}_D)\big)\,.$$

By path induction on p and the characterization of paths in Π-types, this type is equivalent to

$$(\Sigma p : \mathsf{Id}(C, D))(\Pi x : A)(\Pi u : B(x) \to W)\mathsf{Id}\big((\mathsf{ext}\, p)(\mathrm{sup}_C(x, u)), \mathrm{sup}_D(x, (\mathsf{ext}\, p) \circ u)\big)\,,$$

where $\mathsf{ext} : \mathsf{Id}(C, D) \to \mathsf{Equiv}(C, D)$ is the canonical extension function for the identity types of elements of U, asserted to be an equivalence by the univalence axiom. Hence, the above type is equivalent to

$$(\Sigma f : \mathsf{Equiv}(C, D))(\Pi x : A)(\Pi y : B(x) \to W)\, \mathsf{Id}\big(f(\mathrm{sup}_C(x, u)), \mathrm{sup}_D(x, fu)\big)\,.$$

After rearranging, we get

$$(\Sigma f : \mathsf{Alg}((C, \mathrm{sup}_C), (D, \mathrm{sup}_D))\, \mathsf{isequiv}(f)\,.$$

By Proposition 4.11, this type is equivalent to $\mathsf{AlgEquiv}\big((C, \mathrm{sup}_C), (D, \mathrm{sup}_D)\big)$, as desired. Finally, it is not hard to see that the composition of the above equivalences yields, up to a homotopy, the canonical function ext which is therefore an equivalence, as required. □

The following corollary, still obtained under the assumption of the univalence axiom, shows that homotopy-initial algebras are unique up to a unique path.

Corollary 5.16. *Assuming the univalence axiom, homotopy-initial P-algebras are unique up to a contractible type of paths, i.e. the type*

$$(\Pi C : \mathsf{Alg})(\Pi D : \mathsf{Alg})\big(\mathsf{ishinit}(C) \times \mathsf{ishinit}(D) \to \mathsf{iscontr}(\mathsf{Id}(C, D))\big)\,.$$

is inhabited.

Proof. This is an immediate consequence of Theorem 5.15 and Proposition 5.2. □

Acknowledgements

We would like to thank Vladimir Voevodsky and Michael Warren for helpful discussions on the subject of this paper. In particular, the former suggested a simplification of the proof that an inductive P-algebra is homotopy-initial.

Acknowledgements for Steve Awodey. Steve Awodey gratefully acknowledges the support of the National Science Foundation, Grant DMS-1001191 and the Air Force OSR, Grant Number 11NL035 and MURI Grant Number FA9550-15-1-0053.

Acknowledgements for Nicola Gambino.

(i) This work was supported by the National Science Foundation under agreement No. DMS-0635607.

(ii) This material is based on research sponsored by the Air Force Research Laboratory, under agreement number FA8655-13-1-3038.
(iii) This research was supported by a grant from the John Templeton Foundation.
(iv) This research was supported by an EPSRC grant (EP/M01729X/1).

Acknowledgements for Kristina Sojakova. The support of CyLab at Carnegie Mellon under grants DAAD19-02-1-0389 and W911NF-09-1-0273 from the Army Research Office is gratefully acknowledged.

References

[1] M. Abbott, T. Altenkirch, and N. Ghani. Containers: Constructing strictly positive types. *Theoretical Computer Science*, 342(1):3–27, 2005.
[2] P. Aczel. The structure identity principle and the univalence axiom. *The Bullettin of Symbolic Logic*, 342(3):376, 2014.
[3] S. Awodey, N. Gambino, and K. Sojakova. Inductive types in Homotopy Type Theory. In *Logic in Computer Science (LICS 2012)*, pages 95–104. IEEE Computer Society, 2012.
[4] S. Awodey and M. Warren. Homotopy-theoretic models of identity types. *Mathematical Proceedings of the Cambridge Philosophical Society*, 146(1):45–55, 2009.
[5] B. van den Berg and R. Garner. Types are weak ω-groupoids. *Journal of the London Mathematical Society*, 102(2):370–394, 2011.
[6] B. van den Berg and I. Moerdijk. W-types in homotopy type theory. *Mathematical Structures in Computer Science*, FirstView:1–16, 2015.
[7] Y. Bertot and P. Castéran. *Interactive Theorem Proving and Program Development. Coq'Art: the Calculus of Inductive Constructions*. Springer-Verlag, 2004.
[8] R. Blackwell, G. M. Kelly, and A. J. Power. Two-dimensional monad theory. *Journal of Pure and Applied Algebra*, 59(1):1–41, 1989.
[9] M. Boardman and R. Vogt. *Homotopy-invariant algebraic structures on topological spaces*. Number 347 in Lecture Notes in Mathematics. Springer-Verlag, 1973.
[10] T. Coquand and C. Paulin-Mohring. Inductively defined types. In *International Conference on Computer Logic (COLOG '88)*, volume 417 of *LNCS*. Springer, 1990.
[11] P. Dybjer. Representing inductively defined sets by well-orderings in Martin-Löf's type theory. *Theoretical Computer Science*, 176:329–335, 1997.
[12] N. Gambino and R. Garner. The identity type weak factorisation system. *Theoretical Computer Science*, 409(3):94—109, 2008.
[13] N. Gambino and M. Hyland. Well-founded trees and dependent polynomial functors. In S. Berardi, M. Coppo, and F. Damiani, editors, *Types for Proofs and Programs (TYPES '03)*, volume 3085 of *LNCS*, pages 210–225, 2004.
[14] H. Goguen and Z. Luo. Inductive data types: well-ordering types revisited. In G. Huet and G. Plotkin, editors, *Logical Environments*, pages 198–218. Cambridge University Press, 1993.
[15] M. Hofmann. *Extensional constructs in intensional type theory*. Springer-Verlag, 1997.
[16] M. Hofmann and T. Streicher. The groupoid interpretation of type theory. In *Twenty-five years of constructive type theory 1995*, volume 36 of *Oxford Logic Guides*, pages 83–111. Oxford Univ. Press, 1998.
[17] A. Joyal. Categorical homotopy type theory. Slides of a seminar given at MIT. Available from `http://ncatlab.org/homotopytypetheory/files/Joyal.pdf`, 2014.
[18] C. Kapulkin, P. LeFanu, and V. Voevodsky. The simplicial model of univalent foundations. ArXiv:1211.2851, 2014.
[19] C. Kapulkin, P. Lumsdaine, and V. Voevodsky. The simplicial model of univalent foundations. arXiv:1211.2851v1, 2012.
[20] J. Kock. Data types with symmetries and polynomial functors over groupoids, Proceedings of the 28th Conference on the Mathematical Foundations of Programming Semantics, Bath 2012. *Electronic Notes in Theoretical Computer Science*, 286:351–65, 2012.
[21] F.W. Lawvere. An elementary theory of the category of sets. *Proceedings of the National Academy of Sciences*, 52(6):1506–11, 1964.
[22] P. Lumsdaine. *Higher categories from type theories*. PhD thesis, Carnegie Mellon University, 2010.
[23] P. Lumsdaine. Weak ω-categories from intensional type theory. *Logical Methods in Computer Science*, 6:1–19, 2010.
[24] J. Lurie. *Higher topos theory*. Princeton University Press, 2009.

[25] P. Martin-Löf. An intuitionistic theory of types: Predicative part. In H. Rose and J. Shepherdson, editors, *Logic Colloquium 1973*, pages 73—118. North-Holland, 1975.
[26] P. Martin-Löf. Intuitionistic type theory. Notes by G. Sambin of a series of lectures given in Padua, 1980, 1984. Bibliopolis.
[27] I. Moerdijk and E. Palmgren. Well-founded trees in categories. *Annals of Pure and Applied Logic*, 104:189–218, 2000.
[28] B. Nordström, K. Petersson, and J. Smith. *Programming in Martin-Löf type theory*. Oxford University Press, 1990.
[29] B. Nordström, K. Petersson, and J. Smith. Martin-Löf type theory. In *Handbook of Logic in Computer Science*, volume 5, pages 1—37. Oxford University Press, 2000.
[30] T. Streicher. Investigations into intensional type theory, 1993. Habilitation Thesis. Available from the author's web page.
[31] The Univalent Foundations Program, Institute for Advanced Study. *Homotopy Type Theory - Univalent Foundations of Mathematics*. Univalent Foundations Project, 2013.
[32] V. Voevodsky. Notes on type systems, 2009. Available from the author's web page.
[33] V. Voevodsky. An experimental library of formalized mathematics based on the univalent foundations. *Mathematical Structures in Computer Science*, FirstView:1–17, 2015.

Steve Awodey, Department of Philosophy Carnegie Mellon University Pittsburgh, PA 15213, USA
E-mail address: awodey@cmu.edu

Nicola Gambino, School of Mathematics, University of Leeds, Leeds LS2 9JT, UK
E-mail address: n.gambino@leeds.ac.uk

Kristina Sojakova, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA
E-mail address: kristinas@cmu.edu