Natural numbers from integers

***Abstract*—In homotopy type theory, a natural number type is freely generated by an element and an endomorphism. Similarly, an integer type is freely generated by an element and an automorphism. Using only dependent sums, identification types, extensional dependent products, and a type of two elements with large elimination, we construct a natural number type from an integer type. As a corollary, homotopy type theory with only Σ, Id, Π, and finite colimits with descent (and no universes) admits a natural number type. This improves and simplifies a result by Rose and settles a long-standing open question.**

I. Introduction

In set theory and other impredicative background theories, an object of natural numbers is easily constructed from some source of infinity. As soon as we have a set X with a "zero' element $z : X$ and a "successor" embedding $s : X \to X$ whose image does not contain z, we may carve out the natural numbers by taking the intersection of all subsects of X closed under z and s. However, this reasoning is essentially *impredicative*: for this definition to make sense, we need subsets of X to again be a set, that is, the power set of X to exist. In predicative settings, such as Martin-Löf's dependent type theory, this reasoning is ill-founded. Therefore, the natural numbers (and other inductive types) are usually axiomatically assumed.

Homotopy type theory is a version of Martin-Löf's type theory with functional extensionality, univalence for any universes that are assumed, and generalizations of inductive types called *higher inductive types*. A special case of a non-recursive higher inductive types is the *circle* S^1, freely generated by an element b and an identification of b with itself. As proved in [1], the loop space of the circle (the type of self-identifications of its base point) has the universal properties of the *integers*: it is freely generated by an element (given by the trivial self-identification) and an automorphism (given by running along the circle). This object feels infinitary, even though formally speaking is generated using purely finitary means: the higher inductive type of the circle does not have any recursive constructors. Categorically speaking, it arises from only finite limits and colimits.

The loop space of the circle provides a free source of infinity in the higher-dimensional setting. If our theory was impredicative, we could attempt to use to replay the previous argument and construct the natural numbers. This leaves an open question: in a predicative theory such as homotopy type theory, can we construct the natural numbers form the integers? To our knowledge, this question was first raised in a discussion between Egbert Rijke and Mike Shulman on the nForum [2]. (Instead of the integers, we may also assume non-recursive higher inductive types such as the circle and assume a univalent universe.)

The first real progress made towards this question was made by Robert Rose in his PhD thesis [3]. Surprisingly to many experts, he was able to construct the natural numbers from the integers in homotopy type theory. However, his construction is quite complicated, and needs two univalent universes (only the outer one may be replaced by large elimination).

Our contribution in this paper is to provide a new construction that we believe to be much simpler. Furthermore, our construction is more general: we do not need any univalent universes whatsoever. Instead, we just rely on effectivity of finite coproducts (that the coprojections are disjoint embeddings; equivalently, that we have a two-element type that satisfies descent, a homotopically weakened version of large elimination). In fact, we present two versions of our construction: a direct one in Section VI and an indirect one in Section VII.

The categorical analogue of our result is the following. Suppose we have a locally cartesian closed (higher) category with finite coproducts that satisfy descent. Then given an integer object, we have a natural number object. In particular, any locally cartesian closed (higher) category with finite colimits that satisfy descent has a natural number object.

The higher categorical version of our setting is a locally cartesian closed higher category with finite coproducts that satisfy descent and an initial integer algebra. As per Remark 3, this is in particular provided by a locally cartesian closed higher categories with finite colimits that satisfy descent. When replayed in that setting (in the continued absence of an internal language result), the analogue of our constructions in that setting then prove that such a (higher) category has a natural number object.[1]

II. Setting: a restricted type theory

We work in dependent type theory with only a limited set of type formers as set out below. We use $\equiv$ to denote judgmental equality and $\equiv_{\mathsf{def}}$ to denote judgmental definitions. We generally follow the homotopy type theory book [5] for notational conventions.

We have the identification type $x =_A y$ for $x, y : A$ in the sense of Martin-Löf (no equality reflection or axiom K). We have the unit type $\top$, dependent sum types $\sum_{(x:A)} B(x)$, and dependent product types $\prod_{(x:A)} B(x)$, all satisfying judgmental β- and η-laws. Dependent products are

[1] A weaker higher categorical result, namely that every elementary higher topos has a natural number object, is claimed in [4]. Note that an elementary higher topos is impredicative by definition, that is, has access to size-preserving power objects. This simplifies the problem greatly. Furthermore, we believe that the approach of that paper is broken, due to a mistake in the proof of Lemma 1.2.1: $\mathsf{Eq}(g_1, g_2)$ is not generally a subspace of $\mathsf{Map}_{/X}(g_1, g_2)$.

extensional, meaning functional extensionality holds (phrased using identification types).

We assume a *two-element type* $\mathbf{2}$, freely generated by elements $0, 1 : \mathbf{2}$. Its universal property says that for any type C with elements $d_0, d_1 : C$, the type of maps $f : \mathbf{2} \to C$ with identifications $f(0) =_C d_0$ and $f(1) =_C d_1$ is contractible. From this, we can derive a dependent elimination principle whose reduction rule holds up to identification type.

A. Descent

We assume the two-element type satisfies *descent*. This encapsulates elimination into a univalent universe, allowing us to omit universes from our type theory. The principle is as follows. Given types A_0 and A_1, we have a family $C(x)$ over $x : \mathbf{2}$ together with equivalences $C(0) \simeq A_0$ and $C(1) \simeq A_1$. Here, the notion of equivalence is defined as usual in homotopy type theory (using dependent sums and products). This is a homotopically weakened version of large elimination.

B. Basic consequences

a) Elimination principle for two-element type: From the universal property of the two-element type $\mathbf{2}$, we can derive a dependent elimination principle whose reduction rule holds up to identification type. More precisely, given $C(x)$ for $x : \mathbf{2}$ with $d_0 : C(0)$ and $d_1 : C(1)$, we obtain $e(x) : C(x)$ for $x : \mathbf{2}$ together with identifications $e(0) =_C d_0$ and $e(1) =_C d_1$. Moreover, the type of such e is contractible (this uses functional extensionality).

b) Empty type: We may construct the empty type as

$$\bot \equiv_{\mathsf{def}} 0 =_{\mathbf{2}} 1.$$

Its recursion principle follows from descent for the two-element type: given any type A, we have a family $C(x)$ over $x : \mathbf{2}$ with $C(0) \simeq \top$ and $C(1) \simeq A$. From $y : 0 =_{\mathbf{2}} 1$, we then get $C(0) \simeq C(1)$, so $A \simeq \top$, meaning that A is contractible. From this, we obtain the elimination principle for $\bot$ as usual.

c) $\mathbf{2}$ is a set: By descent, we have a family $C(x)$ over $x : \mathbf{2}$ with $C(0) \simeq \top$ and $C(1) \simeq \bot$. The sum over $x : \mathbf{2}$ of $C(x)$ has an element at $x \equiv 0$. By induction over $\mathbf{2}$ and $\bot$, every element in the sum is equal to it. This shows that $\sum_{(x:\mathbf{2})} C(x)$ is contractible. Since $C(0)$ is contractible, it follows that $0 =_{\mathbf{2}} 0$ is contractible. A dual argument shows that $1 =_{\mathbf{2}} 1$ is contractible, making $\mathbf{2}$ a set by $\mathbf{2}$-induction.

d) Binary coproducts: From descent for the two-element type, we can construct the *binary coproduct type* $A_0 \amalg A_1$ of types A_0 and A_1. First, we obtain $C(x)$ over $x : \mathbf{2}$ from descent. We then define

$$A_0 \amalg A_1 \equiv_{\mathsf{def}} \sum_{x:\mathbf{2}} C(x).$$

The coprojections $\tau_i : A_i \to A_0 \amalg A_1$ for $i \in \{0, 1\}$ are defined by passing backward along the equivalence $C(i) \simeq A_i$. The universal property of the binary coproducts reduces using dependent products to the universal property of the two-element type. Given functions $f_i : A_i \to C$ for $i \in \{0, 1\}$, we write $[f_0, f_1] : A \to C$ for the map induced by recursion. Equivalently, the dependent elimination principle for coproducts is justified by that principle for the two-element type. Its reduction rules again hold up to identification type. Given $C(x)$ for $x : \mathbf{2}$ and $f(a_i) : C(\tau_i(a_i))$ for $i \in \{0, 1\}$, our notation for dependent elimination is $[f_0, f_1] : \prod_{(x:\mathbf{2})} C(x)$.

e) No confusion for binary coproducts: The constructors for binary coproducts are disjoint embeddings. Disjointness follows from the tautology $\neg(0 =_{\mathbf{2}} 1)$. The embedding property reduces to contractibility of $0 =_{\mathbf{2}} 0$ and $1 =_{\mathbf{2}} 1$.

f) Descent for binary coproducts: No confusion implies descent for binary products (in fact, it is equivalent to our assumption of descent for the two-element type). This is the following principle, again encapsulating a homotopically weakened version of large elimination into a univalent universe. Given families $C_i(a_i)$ over $a_i : A_i$ for $i \in \{0, 1\}$, we have a family $D(y)$ over $y : A_0 \amalg A_1$ with equivalences $D(\tau_i(a_i)) \simeq C_i(a_i)$. To justify it, we first construct

$$D' \equiv_{\mathsf{def}} \Big(\sum_{a_0:A_0} C_0(a_0)\Big) \amalg \Big(\sum_{a_1:A_1} C_1(a_1)\Big).$$

Coproduct recursion induces a map $p : D' \to A_0 \amalg A_1$. By no confusion, this map pulls back along τ_i for $i \in \{0, 1\}$ to the projection from $\sum_{(a_i:A_i)} C_i(a_o)$ to A_i. We may thus define $D(y)$ as the fiber of p over y.

g) Finite coproducts with no confusion and descent: Of course, we may reduce ternary coproducts and and higher to binary coproducts. The same holds for properties such as no-confusion or descent. For a coproduct of (external) arity $k \in \mathbb{N}$, we denote the coprojections by τ_i where $0 \leq i < k$.

III. Natural numbers and integers

A *natural number algebra* is a type A together with an element $z : A$ and an endofunction $s : A \to A$. Note that this is an external notion (lacking universes, we may not quantify internally over types A in our type theory). We generally refer to a natural number algebra just by its underlying type. We use z and s generically to refer to the structure components of a natural number algebra A. The *structure map* of A is the map $\top \amalg A \to A$ induced by z and s.

Given natural number algebras A and B, the type of *algebra morphisms* from A to B consists of a function $f : A \to B$ with $f(z) =_B z$ and $f(s(a)) =_B s(f(a))$. We call a natural number algebra A *initial* if this type is contractible for any B. We then say A is a *natural number type*. Every natural number algebra A has an identity algebra morphisms id_A. Algebra morphisms $f : A \to B$ and $g : B \to C$ admit a composition $g \circ f : A \to C$. These operations satisfy neutrality and associativity laws (phrased using identification types). We will not need any higher coherence conditions.

We also have displayed analogues of these notions. Given a natural number A, a *natural number algebra B displayed over A* is a family $B(a)$ over $a : A$ with $z : B(z)$ and $s_a(b) : B(s(a))$ for $a : A$ and $b : B(a)$. Its *total algebra* is obtained by taking the dependent sum. The type of *sections* of B consists of $b(a) : B(a)$ for $a : A$ together with $b(z) =_{B(z)} z$

and $b(s(a)) =_{B(s(a))} s(b(a))$ for $a : A$. We say that A has *elimination* if every natural number algebra displayed over it has a section. As is standard, one proves (externally):

Lemma III.1. *A natural number algebra is initial exactly if it has elimination.* □

The following notion features centrally in our constructions.

Definition III.2. *A natural number algebra A is* stable *if its structure map $[z, s] : \top \amalg A \to A$ is an equivalence.*

This means that $z : \top \to A$ and $s : A \to A$ form a colimiting cocone, i.e., that A is *non-recursively* freely generated by z and s. Lambek's lemma states that every initial natural number algebra is stable.

We call a natural number algebra A an *integer algebra* if its endofunction s is an equivalence. Note that this condition is a proposition. This justifies defining morphisms of integer algebras as morphisms of the underlying natural number algebras. An integer algebra A is *initial* if the type of integer algebra morphisms from A to B is contractible for any integer algebra B. We then say that A is an *integer type* (more verbosely, a *type of integers*).

We have a notion of displayed integer algebra analogous to the case of natural number algebras. The type of sections of a displayed integer algebra is defined as the type of sections of the underlying displayed natural number algebra. Analogous to the case of natural numbers, we have:

Lemma III.3. *An integer algebra is initial exactly if it has elimination.* □

Our goal in the rest of this article is to construct a natural number type from an integer type. We thus now make the standing assumption of an integer type $\mathbb{Z}$, with element denoted $Z : \mathbb{Z}$ and automorphism denoted $S : \mathbb{Z} \simeq \mathbb{Z}$ (to distinguish from the natural number algebras we will consider). Note that any other integer type is equivalent to it (by universality). It thus makes sense to speak of *the* integer type $\mathbb{Z}$.

The rest of this paper is devoted to proving the following:

Theorem III.4. *Assume an integer type. Then we have a natural number type.*

We provide two separate proofs, a direct one in Section VI and an indirect one in Section VII.

IV. An equivalence $\mathbb{Z} \simeq \mathbb{Z} \amalg \mathbb{Z}$

Our starting point for constructing the natural numbers is the following observation.

Lemma IV.1. *We have an equivalence $\mathbb{Z} \simeq \mathbb{Z} \amalg \mathbb{Z}$.*

Proof. We first define the operation of *squaring* integer algebras. Given an integer algebra $X \equiv (X, z, s)$, its square $\mathsf{Sq}(X)$ is the integer algebra $(X, z, s \circ s)$. Note that $s \circ s$ is an equivalence since s is. This operation is functorial: it has an evident action on morphisms of integer algebras (we do not need any higher witnesses of functoriality). Furthermore, the functorial action reflects equivalences: if $\mathsf{Sq}(f)$ is invertible, then so is f.

Next, for an integer algebra $X \equiv (X, z, s)$, we define the *twisted rotation* integer algebra $\mathsf{Tw}(X)$ with carrier $X \amalg X$, element $\tau_0(z)$, and automorphism r relating $\tau_0(x)$ with $\tau_1(x)$ and $\tau_1(x)$ with $\tau_0(s(x))$. This uses the universal property of binary coproducts to define r.

Note that the automorphism of $\mathsf{Sq}(\mathsf{Tw}(X))$ relates $\tau_0(x)$ with $\tau_0(s(x))$ and $\tau_1(x)$ with $\tau_1(s(x))$. That is, on each component, it is just given by the original automorphism s. In particular, τ_0 forms an algebra morphism from X to $\mathsf{Sq}(\mathsf{Tw}(X))$.

By initiality of $\mathbb{Z}$, we have an algebra map $\mathbb{Z} \to \mathsf{Sq}(\mathbb{Z})$. Let us call its underlying map $\mathsf{double} : \mathbb{Z} \to \mathbb{Z}$. Note that

$$[\mathsf{double}, S \circ \mathsf{double}] : \mathbb{Z} \amalg \mathbb{Z} \to \mathbb{Z}$$

forms an algebra map from $\mathsf{Tw}(\mathbb{Z})$ to $\mathbb{Z}$.

To construct an equivalence $\mathbb{Z} \simeq \mathbb{Z} \amalg \mathbb{Z}$ of types, it suffices to show that the unique algebra map $c : \mathbb{Z} \to \mathsf{Tw}(\mathbb{Z})$ is invertible. By initiality of $\mathbb{Z}$, we have that $[\mathsf{double}, S \circ \mathsf{double}] \circ c$ is the identity morphism, in particular invertible. If we can show that

$$u \equiv_{\mathsf{def}} c \circ [\mathsf{double}, S \circ \mathsf{double}]$$

is invertible, then c will be invertible by 2-out-of-6 for equivalences. In fact, we will show that u is the identity (on underlying maps).

Note that $\mathsf{Sq}(u)$ is an algebra endomorphism on $\mathsf{Sq}(\mathsf{Tw}(\mathbb{Z}))$. By initiality of $\mathbb{Z}$, we have $\mathsf{Sq}(u) \circ \tau_0 = \tau_0$. On underlying maps, that is $u \circ \tau_0 = \tau_0$. It remains to check $u \circ \tau_1 = \tau_1$. For $x : \mathbb{Z}$, we calculate

$$\begin{aligned} u(\tau_1(x)) &= c(S(\mathsf{double}(x))) \\ &= r(d(\mathsf{double}(x))) \\ &= r(u(\tau_0(x))) \\ &= r(\tau_0(x)) \\ &= \tau_1(x). \end{aligned}$$

□

V. Approximating the integers via halves

If we already had the natural numbers $\mathbb{N}$, we could describe the integers as being built out of two copies of the natural numbers, one for the positive and one for the negative half:

$$\mathbb{Z} \simeq \mathbb{N} \amalg \top \amalg \mathbb{N} \tag{1}$$

Conversely, we may use a decomposition with similar properties to tell us something about the integers, for example when an integer is positive or negative. This is the intuition behind the following construction.

Construction V.1. *Consider types A and B with an equivalence $e : B \amalg A \simeq A$. Then we have an automorphism on $A \amalg B \amalg A$ given by reassociating the equivalence*

$$A \amalg (B \amalg A) \simeq (A \amalg B) \amalg A$$

where we act using e on the left component and using the inverse of e on the left component. Given an element $b : B$, this forms an integer algebra structure on $A \amalg B \simeq A$.

In the above situation, initiality of $\mathbb{Z}$ provides us with an algebra morphism from $\mathbb{Z}$ to $A \amalg B \amalg A$. Restricting along the underlying function, the ternary decomposition on the right induces a decomposition

$$\mathbb{Z} \simeq \mathbb{Z}^- \amalg \mathbb{Z}^0 \amalg \mathbb{Z}^+ \tag{2}$$

of $\mathbb{Z}$ into three parts (this uses effectiveness of coproducts — that the coprojections are disjoint embeddings). The automorphism S of $\mathbb{Z}$ restricts to separate equivalences $S^- : \mathbb{Z}^- \simeq \mathbb{Z}^- \amalg \mathbb{Z}^0$ and $S^+ : \mathbb{Z}^0 \amalg \mathbb{Z}^+ \simeq \mathbb{Z}^+$ that combine to give S via the above decomposition. Furthermore, since $Z : \mathbb{Z}$ is sent to $\tau_1(b)$, we know that Z lies in the middle component $\mathbb{Z}^0$.

In the presence of the naturals, one may prove that the decomposition (2) in fact agrees with the decomposition (1).

VI. First approach: direct

We now give a direct construction of an initial natural number algebra. First we apply Construction V.1 to the equivalence $\mathbb{Z} \simeq \mathbb{Z} \amalg \mathbb{Z}$ from Lemma IV.1 and the element $Z : \mathbb{Z}$ to obtain the decomposition (2). We write M for $\mathbb{Z}^0 \amalg \mathbb{Z}^+$. By construction, for $x : \mathbb{Z}$, we have that x lies in M iff $S(x)$ lies in $\mathbb{Z}^+$, so S restricts to an equivalence $M \simeq \mathbb{Z}^+$. Thus we get also an equivalence $\mathbb{Z}^0 \amalg M \simeq M$. We write S also for the induced map $M \to M$ lying above $S : \mathbb{Z} \to \mathbb{Z}$.

We will show that M essentially has the universal property of the naturals: it is freely generated by $\mathbb{Z}^0 \to M$ and $S : M \to M$. Note also that we have an element of $\mathbb{Z}^0$ – namely Z. To construct $\mathbb{N}$ from here, we will use a simple rectification argument.

We first explain how to prove that M has the stated universal property. We essentially follow a well-known strategy for reducing natural number recursion (which constructs *functions* out of $\mathbb{N}$) to natural number induction (which proves (proposition-valued) *predicates* on $\mathbb{N}$) by considering an appropriate notion of "partially defined inductive function". To this end, we first need a notion of ordering on $\mathbb{Z}$.

Lemma VI.1. *$\mathbb{Z}$ has a proposition-valued relation $<$ with the following properties:*
(i) if $x < y$, then $x < S(y)$,
(ii) if $S(x) < S(y)$, then $x < y$,
(iii) if $x : M$ then we do not have $x < Z$
(iv) $x < S(x)$ for all x.

Proof. Using initiality of $\mathbb{Z}$, we can define subtraction on $\mathbb{Z}$ such that $x - Z = x$ and $x - S(y) = S^{-1}(x - y)$. It can then be proven by integer induction that $S(x) - S(y) = x - y$ and $x - x = Z$. We take $x < y$ to mean that $x - y$ lies in $\mathbb{Z}^-$. All the listed properties can be verified directly. □

We will suppress witnesses of the relation $<$, writing a dash in their place, relying on references to the previous lemma to fill them as required.

We recall some preliminaries on fixpoints. Given an endofunction t on a type X, we write $\mathsf{fix}(t)$ for the type of *fixpoints* of t, defined as

$$\mathsf{fix}(t) \equiv_{\mathsf{def}} \sum_{x:X} t(x) = x.$$

The below "rolling rule" is useful for manipulating fixpoints.

Lemma VI.2 (Rolling rule)**.** *For $f : X \to Y$ and $g : Y \to X$, we have an equivalence $\mathsf{fix}(g \circ f) \simeq \mathsf{fix}(f \circ g)$.*

Proof. Both types arise from

$$\sum_{(x:X)} \sum_{(y:Y)} (f(x) = y) \times (g(y) = x)$$

by contracting singletons. □

Now we prove the universal property of M. Suppose we are given a type family A over M together with

- $z_A : \prod_{(x:\mathbb{Z}^0)} A(x)$,
- $s_A : \prod_{(x:M)} A(x) \to A(S(x))$.

Define a type family pfun over $\mathbb{Z}$ by

$$\mathsf{pfun}(u) \equiv_{\mathsf{def}} \prod_{x:M} (x < u) \to A(x).$$

Thus an element of $\mathsf{pfun}(u)$ is a section of A defined on a finite interval of M. For $u : \mathbb{Z}$, we have a *restriction* map

$$\mathsf{res}_u : \mathsf{pfun}(S(u)) \to \mathsf{pfun}(u)$$

using part (i) of Lemma VI.1 and an *extension* map

$$\mathsf{ext}_u : \mathsf{pfun}(u) \to \mathsf{pfun}(S(u))$$

given by a case distinction using the equivalence $\mathbb{Z}^0 + M \simeq M$:

- $\mathsf{ext}_u(f, x, -) \equiv_{\mathsf{def}} z_A(x)$ for x in $\mathbb{Z}^0$,
- $\mathsf{ext}_u(f, S(x), -) = s_A(f(x, -))$ using the backward direction of part (ii) of Lemma VI.1.

Lemma VI.3. *The operations res and ext commute in the sense that $\mathsf{ext}_u \circ \mathsf{res}_u = \mathsf{res}_{S(u)} \circ \mathsf{ext}_{S(u)}$ for $u : \mathbb{Z}$.*

Proof. For each $f : \mathsf{pfun}(S(u))$ and $x : M$ with $x < u$, we have to prove an equality in $A(x)$. We do a case distinction using the equivalence $\mathbb{Z}^0 \amalg M \simeq M$. Each case is direct by unfolding definitions. □

Now define a type family indfun over $\mathbb{Z}$ by

$$\mathsf{indfun}(u) \equiv_{\mathsf{def}} \mathsf{fix}(\mathsf{res}_u \circ \mathsf{ext}_u).$$

It is direct to see that $\mathsf{res}_u(\mathsf{ext}_u(f)) = f$ if and only f satisfies the recursive equation $f(x, -) = z_A(x)$ for $x : \mathbb{Z}^0$ and $f(S(x), -) = s_A(f(x), -)$ whenever these equations make sense. We defined indfun in this more compact way in order to obtain a simple proof of the following result.

Lemma VI.4. *For all $u : \mathbb{Z}$, we have an element $f(u)$ of $\mathsf{indfun}(u)$.*

Proof. By integer induction. We trivially have an element of $\mathsf{indfun}(Z)$, since there is no $x : M$ with $x < Z$. Moreover, we have

$$\begin{aligned} \mathsf{indfun}(S(u)) &\simeq \mathsf{fix}(\mathsf{res}_{S(u)} \circ \mathsf{ext}_u) \\ &\simeq \mathsf{fix}(\mathsf{ext}_u \circ \mathsf{res}_u) \\ &\simeq \mathsf{fix}(\mathsf{res}_u \circ \mathsf{ext}_u) \\ &\simeq \mathsf{indfun}(u) \end{aligned}$$

by Lemma VI.3 and the rolling rule. □

In fact one can strengthen the above result to the claim that $\mathsf{indfun}(u)$ is contractible – so in particular the proof only uses integer induction for propositions – but we will not need this strengthening.

Lemma VI.5. *We have $g : \Pi_{x:M} A(x)$ with $g(x) = z_A(x)$ for $x : \mathbb{Z}^0$ and $g(S(x)) = s_A(g(x))$.*

Proof. We define $g(x)$ by evaluating $f(S(x))$ at x, using part (iv) of Lemma VI.1. The first equation follows from the fact that $f(S(Z))$ is inductive. The second equation follows from the fact that $f(S(x)) = \mathsf{ext}_x(f(x))$, which we have by construction of f and the proof of the rolling rule. □

This finishes the proof that M has the stated universal property.

Lemma VI.6. *Let X and Y be types with maps $\iota : Y \to X$ and $s : X \to X$. Suppose that X is freely generated by these two maps. If Y has an element, then we have a natural number type.*

Proof. Let $z : Y$ be given. We define a self-map $r : X \to X$ by $r(\iota(y)) = \iota(z)$ and $r(s(x)) = s(r(x))$ using the universal property of X. Let $\mathbb{N} \equiv_{\mathsf{def}} \Sigma_{x:X} r(x) = x$ be the type of fixpoints of r. By Lambek's lemma,[2] the map $Y + X \to X$ is an equivalence, so in particular ι and s are embeddings.We have an equivalence $e_\iota : (z = y) \simeq (r(\iota(y)) = \iota(y))$ since ι is an embedding, and $e_s : (r(x) = x)) \simeq (r(s(x)) = s(x))$ since s is an embedding. Thus we have an element $z_{\mathbb{N}} : \mathbb{N}$ given by $(z, e_\iota(\mathsf{refl}))$. We also have an endomorphism $s_{\mathbb{N}} : \mathbb{N} \to \mathbb{N}$ given by $s_{\mathbb{N}}(x, p) = (s(x), e_s(p)$.

We claim that this makes $\mathbb{N}$ a natural number type. Thus let P be a type family over $\mathbb{N}$ with $z_P : P(z_{\mathbb{N}})$ and $s_P : \Pi_{n:\mathbb{N}} P(n) \to P(s_{\mathbb{N}}(n))$. We construct a term $p : \Pi_{x:X} \Pi_{h:r(x)=x} P(x, h)$ using the universal property of X. We first need to construct, for $y : Y$, a term $\Pi_{h:r(\iota(y))=\iota(y)} P(y, h)$, or equivalently $\Pi_{h:z=y} P(y, e_\iota(h))$. We can define this by path induction using z_P. Then we have to construct for $x : X$ and $f : \Pi_{h:r(x)=x} P(x, h)$ a term $\Pi_{h:r(s(x))=s(x)} P(s(x), h)$, or equivalently $\Pi_{h:r(x)=x} P(s(x), e_s(h))$. Given $h : r(x) = x$, we simply use $s_P(x, h)(f(h))$. It is direct to verify that this defines a section of P as a displayed natural numbers algebra. □

Proof of Theorem III.4. Combining Lemma VI.6 with the universal property of M. □

VII. Second approach: indirect

A *family* is a type A together with a type $B(x)$ for $x : A$. We say that a family is closed under binary coproducts if for $a_0, a_1 : A$, we have $t : A$ with an equivalence $B(t) \simeq B(a_0) \amalg B(a_1)$. Similarly, we define when a family is closed under empty types, unit types, etc. The intuition is that we see A as a universe and B as the associated universe family, sending a code in the universe to an actual type. Note that we do not assume that the family is univalent.

Lemma VII.1. *There is a family (U, El) closed under the unit type and finite coproducts.*

Proof. We take $U \equiv_{\mathsf{def}} \mathbb{Z} \to \mathbb{Z}$ and El as taking fixpoints:

$$\mathsf{El}(f) \equiv_{\mathsf{def}} \sum_{x:\mathbb{Z}} f(x) =_{\mathbb{Z}} x.$$

The unit type is coded by the function constant on Z. For finite coproducts, we make use of the equivalence $\mathbb{Z} \simeq \mathbb{Z} \amalg \mathbb{Z}$ provided Lemma IV.1. Under this equivalence, it suffices to exhibit codes as fixpoints of endofunctions on $\mathbb{Z} \amalg \mathbb{Z}$. Calculating the fixpoints of these endofunctions then makes use of the no-confusion property for binary coproducts.

- The empty type is coded by the endofunction on $\mathbb{Z} \amalg \mathbb{Z}$ swapping the two components. The type of fixpoints of this endofunction is empty.
- The binary coproduct of $f, g : U$ is coded by the endofunction on $\mathbb{Z} \amalg \mathbb{Z}$ that is separately f on the left component and g on the right component. Fixpoints of this endofunction are equivalent to the coproduct of fixpoints of f and fixpoints of g. □

Definition VII.2. *A* counting structure *on a natural number algebra A consists, for $x : A$:*

- *types $C(x)$ and $D(x)$,*
- $\mathsf{min}, \mathsf{max} : D(x)$,
- $\mathsf{low}, \mathsf{upp} : C(x) \to D(x)$

such that $D(x)$ is freely generated separately by min *and* upp *as well as* low *and* max*. We further demand the following:*

- $C(z) \equiv \bot$,
- $C(s(x)) \equiv D(x)$ *for* $x : A$,
- $\mathsf{min} \neq \mathsf{max}$ *in* $D(s(x))$ *for* $x : A$.

Here, free generation equivalently means that the map $\top \amalg C(x) \to D(x)$ induced by min and upp is an equivalence (and similarly for low and max). Note that this may be expressed by a propositional type. We may also omit D entirely from the definition and instead demand an automorphism on $\top \amalg C(x)$. The role of $D(x)$ is to symmetrically split this automorphism via a type "in the middle".

The intuition of Definition VII.2 is that we want to associate to a natural number a set of that cardinality. We could enhance the definition to include the requirement that $C(x)$ is a decidable total order with the rest of the data witnessing a *successor equivalence* between the total orders $\top \star C(x)$ and $C(x) \star \top$ (where $A \star B$ denotes the *join* of orders A and B). However, we do not need this for our development.

A *counting algebra* is a natural number algebra equipped with a counting structure.

Lemma VII.3. *We have a counting algebra.*

Proof. We take as carrier the iterated dependent sum of $c : U$ and $d : U$ and $\mathsf{min}, \mathsf{max} : \mathsf{El}(c)$ and $\mathsf{low}, \mathsf{upp} : \mathsf{El}(c) \to \mathsf{El}(d)$ satisfying conditions as in Definition VII.2. The zero element

[2] In the case at hand, we start from an equivalence $\mathbb{Z}^0 \amalg M \simeq M$ and there is no need to invoke Lambek's lemma.

has c given by the codes for the empty and unit types. The successor of an element of above is given by:

- $c' \equiv_{\mathsf{def}} d$ and d' a code for $\mathsf{El}(d) \amalg \top$,
- $\mathsf{min}' = \tau_0(\mathsf{min})$, $\mathsf{max}' = \tau_1(*)$,
- $\mathsf{low}' = \tau_0$ and upp' defined on $\mathsf{low}(x)$ by $\tau_0(\mathsf{upp}(x))$ and on max by $\tau_1(*)$. □

Corollary VII.4. *Every natural number algebra receives a map from a counting algebra.*

Proof. Take the product of the given natural number algebra with the counting algebra of Lemma VII.3. □

Lemma VII.5. *Let A be a counting algebra. Then A admits the following:*

- *a type $M(x)$ for $x : A$ with:*
  - *$M(z)$ is contractible,*
  - *an equivalence $\mathsf{pair} : M(x) \times A \to M(s(x))$,*
- *for $x : A$, an element $\mathsf{last}_x(m) : \top \amalg A$ for $m : M(x)$ with identifications:*
  - *$\mathsf{last}_z(*) = \tau_0(*)$,*
  - *$\mathsf{last}_{s(x)}(\mathsf{pair}(m, y)) = \tau_1(y)$ for $m : M(x)$ and $y : A$,*
- *an element $\mathsf{rest}_x(m) : M(x)$ for $m : M(x)$ with an identification*

  $$\mathsf{rest}_{s(x)}(\mathsf{pair}(m, y)) =_{M(s(x))} \mathsf{pair}(\mathsf{rest}_x(m), \mathsf{next}_x(m))$$

  where $\mathsf{next}_x(m) \equiv_{\mathsf{def}} [z, s](\mathsf{last}_x(m)) : A$.

Proof. We define $M(x) \equiv_{\mathsf{def}} C(x) \to A$. Note that $M(z)$ is contractible since $C(z) \equiv \bot$. The equivalence

$$\mathsf{pair} : (C(x) \to A) \times A \simeq (C(s(x)) \to A)$$

is induced by $C(s(x)) \equiv D(x)$ and $\mathsf{low} : C(x) \to D(x)$ and $\mathsf{max} : D(x)$.

For the other data, we use that $D(x)$ is freely generated by min and upp. The element $\mathsf{last}_x(m) : \top \amalg A$ is defined by case distinction on $\mathsf{max} : D(x)$:

- on min, we return $\tau_0(*)$,
- on $\mathsf{upp}(y)$ with $y : C(x)$, we return $\tau_1(m(y))$.

The element $\mathsf{rest}_x(m)(c) : A$ for $m : C(x) \to A$ and $c : C(x)$ is defined by case distinction on $\mathsf{low}(c) : D(x)$:

- on min, we return z,
- on $\mathsf{upp}(y)$ with $y : C(x)$, we return $s(y)$.

All the required identifications are direct. □

Lemma VII.6. *Every natural number algebra receives a map from a stable natural number algebra (see Definition III.2).*

Proof. Let A denote the given natural number algebra. Using Corollary VII.4, we may reduce to the setting where A comes equipped with a counting structure. From this, we only need the structure given by Lemma VII.5.

The natural number algebra B we construct over A has carrier given by iterated dependent sum of:

- $x : A$,
- $m : M(x)$,
- $q : x = \mathsf{next}_x(m)$.
- $p : m =_{M(x)} \mathsf{rest}_x(m)$,

This type lies over A using the evident projection to the first component. Given $x : A$, we write $B(x)$ for the remaining three components of the "record type" B.

It remains to construct an equivalence $\top \amalg B \to B$ over $[z, s] : \top \amalg A \to A$. We construct this equivalence in a series of steps.

First note that $B(x)$ arises by contracting k and α in the following iterated dependent sum:

- $k : \top \amalg A$,
- $q : x =_A [z, s](k)$,
- $m : M(x)$,
- $\alpha : k =_{\top \amalg A} \mathsf{last}_x(m)$,
- $p : m =_{M(x)} \mathsf{rest}_x(m)$.

Contracting x with q, we have for $k : \top \amalg A$ that $B([z, s](k))$ is equivalent to:

- $m : M(x)$,
- $\alpha : k = \mathsf{last}_x(m)$,
- $p : m = \mathsf{rest}_x(m)$

where $x \equiv_{\mathsf{def}} [z, s](k)$. We transform this type under the cases of k.

For $k = \tau_0(*)$, we have $x = z$ and are left with:

- $m : M(z)$,
- $\alpha : \tau_0(*) = \mathsf{last}_z(m)$,
- $p : m = \mathsf{rest}_z(m)$.

Since $M(z)$ is contractible, this is equivalent to $\tau_0(*) =_{\top \amalg A} \tau_0(*)$. By no-confusion, τ_0 is an embedding, so this is contractible.

For $k = \tau_1(y)$, we have $x = s(y)$ and are left with:

- $m : M(s(y))$,
- $\alpha : \tau_1(y) = \mathsf{last}_{s(y)}(m)$,
- $p : m = \mathsf{rest}_{s(y)}(m)$.

Expanding the product $M(s(y)) \simeq M(y) \times A$, this is equivalent to:

- $m' : M(y)$,
- $z : A$,
- $\alpha : \tau_1(y) = \mathsf{last}_{s(y)}(\mathsf{pair}(m', z))$,
- $p : \mathsf{pair}(m', z) = \mathsf{rest}_{s(y)}(\mathsf{pair}(m', z))$.

This rewrites to:

- $m' : M(y)$,
- $z : A$,
- $\alpha : \tau_1(y) = \tau_1(z)$,
- $p : \mathsf{pair}(m', z) = \mathsf{pair}(\mathsf{rest}_y(m'), \mathsf{next}_y(m'))$.

Since τ_1 is an embedding by no-confusion, we may contract z with α:

- $m' : M(y)$,
- $p : \mathsf{pair}(m', y) = \mathsf{pair}(\mathsf{rest}_y(m'), \mathsf{next}_y(m'))$.

Splitting p into a pair of equalities, we recover $B(y)$. □

The strategy of the above proof is reminiscient of the proof of the rolling rule (Lemma VI.2). In particular, the definition of B is almost that of the fixpoints of an operation on $\sum_{(x:A)} M(x)$. However, type of p seems to resist this (it

is an identification in $M(x)$, not $M(\mathsf{next}_x(m))$). It is unclear to us if this analogy can be exploited further.

Lemma VII.7. *We have a stable natural number algebra that embeds into $\mathbb{Z}$.*

Proof. By Lemma VII.6, we have a stable natural number algebra A with a morphism $f : A \to \mathbb{Z}$. We now apply Construction V.1 to the type $A \equiv_{\mathsf{def}} A$ and $B \equiv_{\mathsf{def}} \top$. The equivalence $A \simeq B \amalg A$ is just the structure map of the stable natural number algebra A. Note that B has a unique element. The resulting integer algebra (with carrier $A \amalg \top \amalg A$) again lies over $\mathbb{Z}$: we send $\tau_0(a)$ to $S^{-1}(\mathsf{inv}(a))$, $\tau_1(*)$ to Z, and $\tau_2(a)$ to $S(f(a))$. Here, inv is the underlying map of the unique integer algebra morphism from $\mathbb{Z}$ to $\mathbb{Z}'$ where $\mathbb{Z}'$ has automorphism S^{-1} instead of S.

By initiality of $\mathbb{Z}$, the algebra map $\mathbb{Z} \to A \amalg \top \amalg A$ is a section of the algebra map $A \amalg \top \amalg A \to \mathbb{Z}$. By construction, the former map sends $\mathbb{Z}^0$ in the decomposition (2) to the middle component in $A \amalg \top \amalg A$, and in turn, that middle component is sent to $\mathbb{Z}^0$ by the latter map (specifically, to Z). This exhibits Z^0 as a retract of $\top$, in particular it is contractible. We may thus silently replace Z^0 by $\top$ in the obtained decomposition (2).

This makes $\top \amalg \mathbb{Z}^+$ with zero element $\tau_0(*)$ and successor function $\tau_1 \circ S^+$ into a stable natural number algebra embedding into $\mathbb{Z}$. □

In the last step of the above lemma, we can equivalently directly use $\mathbb{Z}^+$ as the desired natural number algebra. Denote the image of the zero in $\mathbb{Z}$ by t. We then need to postcompose with the shifting map of $\mathbb{Z}$ that sends t to Z to obtain the algebra embedding from $\mathbb{Z}^+$ to $\mathbb{Z}$.

Lemma VII.8. *Let A be stable natural number algebra over $\mathbb{Z}$ such that $A(Z)$ is contractible. Then A is initial.*

Proof. Using Lemma III.3, it suffices to show that A has elimination. Note that, in natural number algebras over A, having a section is a covariant structure. That is, given a morphism $E' \to E$ of natural number algebras over A, if E' has a section, then so does E. Using Lemma VII.6, it thus suffices to construct a section for a *stable* natural number algebra E over A. In fact, we will show that E is fiberwise contractible over A.

Since both A and E are stable, we have $E(z) \simeq 1$ and $E(s(a)) \simeq E(a)$.

We define an integer algebra Q over $\mathbb{Z}$ that is a family of propositions by setting

$$Q(x) \equiv_{\mathsf{def}} \prod_{a:A(x)} \mathsf{isContr}(E(a))$$

for $x : A$. Note that $Q(0)$ holds because $z : 1 \to A(Z)$ is an equivalence and $E(z)$ is contractible. It remains to show that $Q(x)$ and $Q(S(x))$ are logically equivalent for $x : \mathbb{Z}$.

From $s : A(x) \to A(S(x))$ and $E(s(a)) \simeq E(a)$, we have $Q(S(x)) \to Q(x)$. Let us check $Q(x) \to Q(S(x))$. Given $f : \prod_{(a:A(x))} \mathsf{isContr}(E(a))$ and $a' : A(S(x))$, we need $\mathsf{isContr}(E(a'))$. We case split on $[z, s]^{-1}(S(x), a') : \top \amalg A$:

- For $(S(x), a') = (0, z)$, the goal follows from $E(z) \simeq 1$.
- For $(S(x), a') = (S(y), s(a))$ where $y : \mathbb{Z}$ and $a : A(y)$, the goal reduces to $\mathsf{isContr}(E(s(a)))$. This follows from $E(s(a)) \simeq E(a)$ and $f(a) : \mathsf{isContr}(E(a))$.

By initiality of $\mathbb{Z}$, we obtain a section q of Q. So, given $a : A(x)$, we have $q(x, a) : \mathsf{isContr}(E(a))$. □

Proof of Theorem III.4. Apply Lemma VII.8 to Lemma VII.7. □

References

[1] N. Kraus and J. von Raumer, "Path spaces of higher inductive types in homotopy type theory," in *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. IEEE, 2019, pp. 1–13.

[2] E. Rijke, Discussion post on the nForum, 2017. [Online]. Available: https://nforum.ncatlab.org/discussion/6691/higher-inductive-type/?Focus=61552#Comment_61552

[3] R. Rose, "The natural numbers in predicative univalent type theory," Ph.D. dissertation, Indiana University, 2020.

[4] N. Rasekh, "Every elementary higher topos has a natural number object," *Theory and Applications of Categories*, vol. 37, no. 13, pp. 337–377, 2021.

[5] T. Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study: https://homotopytypetheory.org/book, 2013.