natural numbers type

Contents

1. Idea
2. Definition
   - With typal computation and uniqueness rules
   - Generalized induction principle
   - Extensionality principle of the natural numbers
3. Properties
   - General
   - Relation to the type of finite types
   - Categorical semantics
     - Recursion
     - Induction
4. Related concepts
5. References

1. Idea

In type theory: the the *natural numbers type* is the type of natural numbers.

2. Definition

Definition 2.1 . The type of natural numbers $\mathbb{N}$ is the inductive type defined by the following inference rules.

1. ***type formation rule***:

$$\frac{}{\mathbb{N} : \mathrm{Type}}$$

2. ***term introduction rules***:

$$\frac{}{0 : \mathbb{N}} \qquad \frac{n : \mathbb{N}}{\mathrm{succ}(n) : \mathbb{N}}$$

3. ***term elimination rule***:

$$\frac{n : \mathbb{N} \vdash P(n) : \mathrm{Type} \; ; \quad \vdash 0_P : P(0) \; ; \quad n : \mathbb{N},\, p : P(x) \vdash \mathrm{succ}_P(n,\, p) : P}{n : \mathbb{N} \vdash \mathrm{ind}_{(P, 0_P, \mathrm{succ}_P)}(n) : P(n)}$$

4. ***computation rules***:

$$\frac{n : \mathbb{N} \vdash P(n) : \mathrm{Type} \; ; \quad \vdash 0_P : P(0) \; ; \quad n : \mathbb{N},\, p : P(x) \vdash \mathrm{succ}_P(n, p) : P(}{\mathrm{ind}_{(P, 0_P, \mathrm{succ}_P)}(0) = 0_P}$$

and

$$\frac{n : \mathbb{N} \vdash P(x) : \mathrm{Type} \; ; \quad \vdash 0_P : P(0) \; ; \quad n : \mathbb{N},\, p : P(x) \vdash \mathrm{succ}_P(x, p) : P(}{\mathrm{ind}_{(P, 0_P, \mathrm{succ}_P)}(\mathrm{succ}(n)) = \mathrm{succ}_P\big(n,\, \mathrm{ind}_{(P, 0_P, \mathrm{succ}_P)}(n)\big)}$$

(In the last line, “=” denotes judgemental equality.)

That this is the right definition (and a special case of the general principle of inductive types) was clearly understood around Martin-Löf (1984), pp. 38; Coquand & Paulin (1990, p. 52-53); Paulin-Mohring (1993, §1.3); Dybjer (1994, §3). For review see also, e.g., Pfenning (2009, §2); UFP (2013, §1.9); Söhnen (2018, §2.4.5).

In Coq-syntax the natural numbers are the inductive type defined [cf. Paulin-Mohring (2014, p. 6)] by:

```
Inductive nat : Type :=
 | zero : nat
 | succ : nat -> nat.
```

In the categorical semantics (via the categorical model of dependent types, see below) this is interpreted as the initial algebra for the endofunctor F that sends an object to its coproduct with the terminal object

$$F(X) \;\coloneqq\; * \sqcup X \,, \tag{1}$$

or in different equivalent notation, which is very suggestive here:

$$F(X) \;=\; 1 + X \;.$$

That initial algebra is (as also explained there) precisely a natural number object $\mathbb{N}$. The two components of the morphism $F(\mathbb{N}) \to \mathbb{N}$ that defines the algebra structure are the 0-element $0 : \; * \; \to \mathbb{N}$ and the successor endomorphism $\mathrm{succ} : \mathbb{N} \to \mathbb{N}$

$$(0, \, \mathrm{succ}) \;:\; * \sqcup \mathbb{N} \longrightarrow \mathbb{N} \;.$$

With typal computation and uniqueness rules

Assuming that identification types, function types and dependent sequence types exist in the type theory, the natural numbers type is the inductive type generated by an element and a function:

Formation rules for the natural numbers type:

$$\frac{\Gamma \; \mathrm{ctx}}{\Gamma \vdash \mathbb{N} \; \mathrm{type}}$$

Introduction rules for the natural numbers type:

$$\frac{\Gamma \; \mathrm{ctx}}{\Gamma \vdash 0 \colon \mathbb{N}} \qquad \frac{\Gamma \; \mathrm{ctx}}{\Gamma \vdash s \colon \mathbb{N} \to \mathbb{N}}$$

Elimination rules for the natural numbers type:

$$\frac{\Gamma, x \colon \mathbb{N} \vdash C(x) \; \mathrm{type} \quad \Gamma \vdash c_0 \colon C(0) \quad \Gamma \vdash c_s \colon \prod_{x:\mathbb{N}} C(x) \to C(s(x)) \quad \Gamma \vdash n \colon \mathbb{N}}{\Gamma \vdash \mathrm{ind}_{\mathbb{N}}^{C}(c_0, c_s, n) \colon C(n)}$$

Computation rules for the natural numbers type:

$$\frac{\Gamma, x \colon \mathbb{N} \vdash C(x) \; \mathrm{type} \quad \Gamma \vdash c_0 \colon C(0) \quad \Gamma \vdash c_s \colon \prod_{x:\mathbb{N}} C(x) \to C(s(x))}{\Gamma \vdash \beta_{\mathbb{N}}^{0}(c_0, c_s) \colon \mathrm{Id}_{C(0)} \mathrm{ind}_{\mathbb{N}}^{C}(c_0, c_s, 0), c_0)}$$

$$\frac{\Gamma, x \colon \mathbb{N} \vdash C(x) \; \mathrm{type} \quad \Gamma \vdash c_0 \colon C(0) \quad \Gamma \vdash c_s \colon \prod_{x:\mathbb{N}} C(x) \to C(s(x)) \quad \Gamma \vdash n \colon \mathbb{N}}{\Gamma \vdash \beta_{\mathbb{N}}^{s}(c_0, c_s, n) \colon \mathrm{Id}_{C(s(n))}(\mathrm{ind}_{\mathbb{N}}^{C}(c_0, c_s, s(n)), c_s(n)(\mathrm{ind}_{\mathbb{N}}^{C}(c_0, c_s, n)))}$$

Uniqueness rules for the natural numbers type:

$$\frac{\Gamma, x:\mathbb{N} \vdash C(x)\ \mathrm{type} \quad \Gamma \vdash c: \prod_{x:\mathbb{N}} C(x) \quad \Gamma \vdash n:\mathbb{N}}{\Gamma \vdash \eta_{\mathbb{N}}(c, n): \mathrm{Id}_{C(n)}(\mathrm{ind}_{\mathbb{N}}^{C}(c(0), \lambda x:\mathbb{N}.c(s(x)), n), c(n))}$$

The elimination, computation, and uniqueness rules for the natural numbers type state that the natural numbers type satisfy the ***dependent universal property of the natural numbers***. If the dependent type theory also has dependent sum types and product types, allowing one to define the uniqueness quantifier, the dependent universal property of the natural numbers could be simplified to the following rule:

$$\frac{\Gamma, x:\mathbb{N} \vdash C(x)\ \mathrm{type} \quad \Gamma \vdash c_0: C(0) \quad \Gamma \vdash c_s: \prod_{x:\mathbb{N}} C(x) \to C(s(x))}{\Gamma \vdash \mathrm{up}_{\mathbb{N}}^{C}(c_0, c_s): \exists! c: \prod_{x:\mathbb{N}} C(x).\mathrm{Id}_{C(0)}(c(0), c_0) \times \prod_{x:\mathbb{N}} \mathrm{Id}_{C(s(x))}(c(s(x)), c_s(c(x)))}$$

The dependent universal property of the natural numbers is used to characterize the dependent product type of an type family $C(x)$ dependent on $x:\mathbb{N}$, and states that the fibers of the function

$$\lambda c.(c(0), \lambda x.c(s(x))): \prod_{x:\mathbb{N}} C(x) \to \left(C(0) \times \prod_{x:\mathbb{N}} C(x) \to C(s(x))\right)$$

are contractible types. This is equivalent to saying that the above function is an equivalence of types:

$$\mathrm{isEquiv}(\lambda c.(c(0), \lambda x.c(s(x))))$$

The non-dependent universal property similarly says that given a type C the function

$$\lambda c.(c(0), \lambda x.c(s(x))): (\mathbb{N} \to C) \to (C \times (\mathbb{N} \to C \to C))$$

is an equivalence of types

$$\mathrm{isEquiv}(\lambda c.(c(0), \lambda x.c(s(x))))$$

Generalized induction principle

There is also a generalized induction principle (cf. the talk slides in LumsdaineShulman17), which uses a type C and a function $f: C \to \mathbb{N}$ instead of a type family $x:\mathbb{N} \vdash P(x)$, and one uses the fiber $\sum_{z:C} f(z) =_{\mathbb{N}} n$ to express the generalized induction principle.

Then the induction principle states that given a type C and a function $f: C \to \mathbb{N}$ along with

- dependent pair

$$c_0 \colon \sum_{z:C} f(z) =_{\mathbb{N}} 0$$

- dependent function

$$c_s \colon \prod_{n:\mathbb{N}} \Big(\sum_{z:C} f(z) =_{\mathbb{N}} n \Big) \to \Big(\sum_{z:C} f(z) =_{\mathbb{N}} s(n) \Big)$$

- and natural number $n \colon \mathbb{N}$

one could construct the dependent pair

$$\mathrm{ind}_{\mathbb{N}}^{C}(f, c_0, c_s, n) \colon \sum_{z:C} f(z) =_{\mathbb{N}} n$$

such that

$$\mathrm{ind}_{\mathbb{N}}^{C}(f, c_0, c_s, 0) \equiv c_0$$

and for all $n \colon \mathbb{N}$

$$\mathrm{ind}_{\mathbb{N}}^{C}(f, c_0, c_s, s(n)) \equiv c_s(n, \mathrm{ind}(f, c_0, c_s, n))$$

However, by the rules of dependent pair types, one could instead postulate separate elements and identifications instead of an element of a fiber type throughout the generalized principle.

Instead of the dependent pair $c_0 \colon \sum_{z:C} f(z) =_{\mathbb{N}} 0$ we have the element $c_0 \colon C$ and identificaiton $p_0 \colon f(c_0) =_{\mathbb{N}} 0$, where the original element is given by (c_0, p_0). In addition, given the dependent type

$$c_s \colon \prod_{n:\mathbb{N}} \Big(\sum_{z:C} f(z) =_{\mathbb{N}} n \Big) \to \Big(\sum_{z:C} f(z) =_{\mathbb{N}} s(n) \Big)$$

by currying this is equivalent to

$$c_s \colon \prod_{n:\mathbb{N}} \prod_{z:C} \big(f(z) =_{\mathbb{N}} n \big) \to \Big(\sum_{z:C} f(z) =_{\mathbb{N}} s(n) \Big)$$

and by the type theoretic axiom of choice this is equivalent to

$$c_s \colon \prod_{n:\mathbb{N}} \prod_{y:C} \sum_{g:(f(y) =_{\mathbb{N}} n) \to C} \prod_{p:f(y) =_{\mathbb{N}} n} f(g(p)) =_{\mathbb{N}} s(n)$$

By the rules of dependent pair types, the family of dependent pair types could be split up into

$$c_s: \prod_{n:\mathbb{N}} \prod_{y:C} (f(y) =_{\mathbb{N}} n) \to C$$

$$p_s: \prod_{n:\mathbb{N}} \prod_{y:C} \prod_{p:f(y) =_{\mathbb{N}} n} f(c_s(n, y, p)) =_{\mathbb{N}} s(n)$$

where the original dependent funciton is given by

$$\lambda n:\mathbb{N}.\lambda y:C.(c_s(n, y), p_s(n, y))$$

Then the induction principle of the natural numbers states that given a type C and a function $f:C \to \mathbb{N}$, along with

- an element $c_0:C$
- an identification $p_0: f(c_0) =_{\mathbb{N}} 0$
- dependent functions

$$c_s: \prod_{n:\mathbb{N}} \prod_{y:C} (f(y) =_{\mathbb{N}} n) \to C$$

$$p_s: \prod_{n:\mathbb{N}} \prod_{y:C} \prod_{p:f(y) =_{\mathbb{N}} n} f(c_s(n, y, p)) =_{\mathbb{N}} s(n)$$

we have a function

$$\mathrm{ind}_{\mathbb{N}}^{C}(f, c_0, p_0, c_s, p_s): \mathbb{N} \to C$$

and a homotopy

$$\mathrm{ind}_{\mathbb{N}}^{C,\mathrm{sec}}(f, c_0, p_0, c_s, p_s): \prod_{n:\mathbb{N}} f(\mathrm{ind}_{\mathbb{N}}^{C}(f, c_0, p_0, c_s, p_s, n)) =_{\mathbb{N}} n$$

indicating that $\mathrm{ind}_{\mathbb{N}}^{C}(f, c_0, p_0, c_s, p_s)$ is a section of f, such that

$$\mathrm{ind}_{\mathbb{N}}^{C}(f, c_0, p_0, c_s, p_s, 0) \equiv c_0$$

$$\mathrm{ind}_{\mathbb{N}}^{C,\mathrm{sec}}(f, c_0, p_0, c_s, p_s, 0) \equiv p_0$$

and for all $n:\mathbb{N}$,

$$\mathrm{ind}_{\mathbb{N}}^{C}(f, c_0, p_0, c_s, p_s, s(n)) \equiv c_s(n, \mathrm{ind}_{\mathbb{N}}^{C}(f, c_0, p_0, c_s, p_s, n), \mathrm{ind}_{\mathbb{N}}^{C,\mathrm{sec}}(f, c_0, p_0, c_s, p_s, n))$$

$$\mathrm{ind}_{\mathbb{N}}^{C,\mathrm{sec}}(f, c_0, p_0, c_s, p_s, s(n)) \equiv p_s(n, \mathrm{ind}_{\mathbb{N}}^{C}(f, c_0, p_0, c_s, p_s, n), \mathrm{ind}_{\mathbb{N}}^{C,\mathrm{sec}}(f, c_0, p_0, c_s, p_s$$

As inference rules these are given by the following:

elimination rules:

$$\frac{\begin{array}{c}\Gamma \vdash C \; \mathrm{type} \quad \Gamma \vdash f: C \to \mathbb{N} \quad \Gamma \vdash c_0: C \quad \Gamma \vdash p_0: f(c_0) =_{\mathbb{N}} 0 \\ \Gamma \vdash c_s: \prod_{n:\mathbb{N}} \prod_{y:C} (f(y) =_{\mathbb{N}} n) \to C \quad \Gamma \vdash p_s: \prod_{n:\mathbb{N}} \prod_{y:C} \prod_{p:f(y) =_{\mathbb{N}} n} f(c_s(n, y, p\end{array}}{\Gamma \vdash \mathrm{ind}_{\mathbb{N}}^{C}(f, c_0, p_0, c_s, p_s): \mathbb{N} \to C}$$

$$\frac{\begin{array}{c}\Gamma \vdash C \; \mathrm{type} \quad \Gamma \vdash f: C \to \mathbb{N} \quad \Gamma \vdash c_0: C \quad \Gamma \vdash p_0: f(c_0) =_{\mathbb{N}} 0 \\ \Gamma \vdash c_s: \prod_{n:\mathbb{N}} \prod_{y:C} (f(y) =_{\mathbb{N}} n) \to C \quad \Gamma \vdash p_s: \prod_{n:\mathbb{N}} \prod_{y:C} \prod_{p:f(y) =_{\mathbb{N}} n} f(c_s(n, y, p\end{array}}{\Gamma \vdash \mathrm{ind}_{\mathbb{N}}^{C,\mathrm{sec}}(f, c_0, p_0, c_s, p_s): \prod_{n:\mathbb{N}} f(\mathrm{ind}_{\mathbb{N}}^{C}(f, c_0, p_0, c_s, p_s, n)) =_{\mathbb{N}} n}$$

computation rules:

$$\frac{\begin{array}{c}\Gamma \vdash C \; \mathrm{type} \quad \Gamma \vdash f: C \to \mathbb{N} \quad \Gamma \vdash c_0: C \quad \Gamma \vdash p_0: f(c_0) =_{\mathbb{N}} 0 \\ \Gamma \vdash c_s: \prod_{n:\mathbb{N}} \prod_{y:C} (f(y) =_{\mathbb{N}} n) \to C \quad \Gamma \vdash p_s: \prod_{n:\mathbb{N}} \prod_{y:C} \prod_{p:f(y) =_{\mathbb{N}} n} f(c_s(n, y, p\end{array}}{\Gamma \vdash \mathrm{ind}_{\mathbb{N}}^{C}(f, c_0, p_0, c_s, p_s, 0) \equiv c_0}$$

$$\frac{\begin{array}{c}\Gamma \vdash C \; \mathrm{type} \quad \Gamma \vdash f: C \to \mathbb{N} \quad \Gamma \vdash c_0: C \quad \Gamma \vdash p_0: f(c_0) =_{\mathbb{N}} 0 \\ \Gamma \vdash c_s: \prod_{n:\mathbb{N}} \prod_{y:C} (f(y) =_{\mathbb{N}} n) \to C \quad \Gamma \vdash p_s: \prod_{n:\mathbb{N}} \prod_{y:C} \prod_{p:f(y) =_{\mathbb{N}} n} f(c_s(n, y, p\end{array}}{\Gamma \vdash \mathrm{ind}_{\mathbb{N}}^{C,\mathrm{sec}}(f, c_0, p_0, c_s, p_s, 0) \equiv p_0}$$

$$\frac{\begin{array}{c}\Gamma \vdash C \; \mathrm{type} \quad \Gamma \vdash f: C \to \mathbb{N} \quad \Gamma \vdash c_0: C \quad \Gamma \vdash p_0: f(c_0) =_{\mathbb{N}} 0 \\ \Gamma \vdash c_s: \prod_{n:\mathbb{N}} \prod_{y:C} (f(y) =_{\mathbb{N}} n) \to C \quad \Gamma \vdash p_s: \prod_{n:\mathbb{N}} \prod_{y:C} \prod_{p:f(y) =_{\mathbb{N}} n} f(c_s(n, y, p\end{array}}{\Gamma \vdash \mathrm{ind}_{\mathbb{N}}^{C}(f, c_0, p_0, c_s, p_s, s(n)) \equiv c_s(n, \mathrm{ind}_{\mathbb{N}}^{C}(f, c_0, p_0, c_s, p_s, n), \mathrm{ind}_{\mathbb{N}}^{C,\mathrm{sec}}(f, c_0,}$$

$$\frac{\begin{array}{c}\Gamma \vdash C \; \mathrm{type} \quad \Gamma \vdash f: C \to \mathbb{N} \quad \Gamma \vdash c_0: C \quad \Gamma \vdash p_0: f(c_0) =_{\mathbb{N}} 0 \\ \Gamma \vdash c_s: \prod_{n:\mathbb{N}} \prod_{y:C} (f(y) =_{\mathbb{N}} n) \to C \quad \Gamma \vdash p_s: \prod_{n:\mathbb{N}} \prod_{y:C} \prod_{p:f(y) =_{\mathbb{N}} n} f(c_s(n, y, p\end{array}}{\Gamma \vdash \mathrm{ind}_{\mathbb{N}}^{C,\mathrm{sec}}(f, c_0, p_0, c_s, p_s, s(n)) \equiv p_s(n, \mathrm{ind}_{\mathbb{N}}^{C}(f, c_0, p_0, c_s, p_s, n), \mathrm{ind}_{\mathbb{N}}^{C,\mathrm{sec}}(f, c}$$

One gets back the usual induction principle of the natural numbers type when $C \equiv \sum_{n:\mathbb{N}} P(n)$ and $f \equiv \pi_1$ the first projection function of the dependent sum type, and one gets back the recursion principle of the natural numbers type when $C \equiv \mathbb{N} \times X$ and $f \equiv \pi_1$ the first projection function of the product type.

Extensionality principle of the natural numbers

First we inductively define a binary function into the boolean domain called observational equality of the natural numbers:

$$\frac{\Gamma\ \mathrm{ctx}}{\Gamma \vdash \mathrm{Eq}_{\mathbb{N}} : \mathbb{N} \times \mathbb{N} \to \mathrm{Bit}}$$

$$\frac{\Gamma\ \mathrm{ctx}}{\Gamma \vdash \delta^{0,0} : \mathrm{Id}_{\mathrm{Bit}}(\mathrm{Eq}_{\mathbb{N}}(0,0), 1)} \qquad \frac{\Gamma\ \mathrm{ctx}}{\Gamma, n:\mathbb{N} \vdash \delta^{0,s}(n) : \mathrm{Id}_{\mathrm{Bit}}(\mathrm{Eq}_{\mathbb{N}}(0, s(n)), 0)}$$

$$\frac{\Gamma\ \mathrm{ctx}}{\Gamma, m:\mathbb{N} \vdash \delta^{s,0}(m) : \mathrm{Id}_{\mathrm{Bit}}(\mathrm{Eq}_{\mathbb{N}}(s(m), 0), 0)} \qquad \frac{\Gamma\ \mathrm{ctx}}{\Gamma, m:\mathbb{N}, n:\mathbb{N} \vdash \delta^{s,s}(m,n) : \mathrm{Id}_{\mathrm{Bit}}(\mathrm{Eq}_{\mathbb{N}}(s(r$$

The extensionality principle of the natural numbers states that the natural numbers has decidable equality given by observational equality:

$$\frac{\Gamma\ \mathrm{ctx}}{\Gamma, m:\mathbb{N}, n:\mathbb{N} \vdash \delta(m,n) : \mathrm{Id}_{\mathbb{N}}(m,n) \simeq \mathrm{El}_{\mathrm{Bit}}(\mathrm{Eq}_{\mathbb{N}}(m,n))}$$

or equivalently

$$\frac{\Gamma\ \mathrm{ctx}}{\Gamma, m:\mathbb{N}, n:\mathbb{N} \vdash \delta(m,n) : \mathrm{Id}_{\mathbb{N}}(m,n) \simeq \mathrm{Id}_{2}(\mathrm{Eq}_{\mathbb{N}}(m,n), 1)}$$

3. Properties

General

Example 3.1. ***(natural numbers type as a $\mathcal{W}$-type)***

The natural numbers type $(\mathbb{N}, 0, \mathrm{succ})$ (Def. 2.1) is equivalently the $\mathcal{W}$-type $\underset{c:C}{\mathcal{W}} A(c)$ with:

- $C \coloneqq \{0, \mathrm{succ}\} \simeq * \sqcup *$;
- $A_0 \coloneqq \varnothing$ (empty type);

$$A_{\mathrm{succ}} \coloneqq * \text{ (unit type)}$$

[Martin-Löf (1984), pp. 45, Dybjer (1997, p. 330, 333)]

Relation to the type of finite types

The natural numbers type is equivalent to the set truncation of the type of finite types:

$$\mathbb{N} \simeq [\mathrm{FinType}]_0$$

This is the type theoretic analogue of the decategorification of the permutation category resulting in the set of natural numbers.

This gives us an alternate definition of the natural numbers as the type of finite types

$$\mathbb{N} \coloneqq [\mathrm{FinType}]_0$$

One has $[-]_0 : \mathrm{FinType} \to [\mathrm{FinType}]_0$ by the introduction rules of set truncation.

The arithmetic operations and order relations on the natural numbers type can be defined by induction on set truncation:

For all finite types $A : \mathrm{FinType}$ and $B : \mathrm{FinType}$ and finite families $C : A \to \mathrm{FinType}$, we have

$$0 =_{\mathbb{N}} [\varnothing]_0 \quad 1 =_{\mathbb{N}} [\mathbb{1}]_0$$

$$[A]_0 + [B]_0 =_{\mathbb{N}} [A + B]_0$$

$$\sum_{x=1}^{[A]_0} [C]_0(x) =_{\mathbb{N}} \Big[\sum_{x:A} C(x)\Big]_0$$

$$[A]_0 \cdot [B]_0 =_{\mathbb{N}} [A \times B]_0$$

$$\prod_{x=1}^{[A]_0} [C]_0(x) =_{\mathbb{N}} \Big[\prod_{x:A} C(x)\Big]_0$$

$$[B]_0^{[A]_0} =_{\mathbb{N}} [A \to B]_0$$

$$[A]_0 =_{\mathbb{N}} [B]_0 \coloneqq [A \simeq B]_{(-1)} \text{ or } [A =_{\mathrm{FinType}} B]_{(-1)}$$

$$[A]_0 \leq [B]_0 \coloneqq [A \hookrightarrow B]_{(-1)}$$

Categorical semantics

We spell out how under the canonical categorical model of dependent types, the categorical semantics of the natural numbers types yields a natural numbers object together with its expected recursion and induction principle.

Throughout, we consider an ambient category $\mathcal{C}$ (e.g. $\mathcal{C} =$ Set) and write

$$F\,\mathrm{Alg}(\mathcal{C}) \xrightarrow{\text{underlying}} \mathcal{C} \qquad (2)$$

for the category of algebras over the endofunctor F (1).

Recursion

We spell out how the fact that $\mathbb{N}$ satisfies Def. 2.1 is the classical recursion principle.

We begin with a simple special case of recursion (cf. Rem. 3.2), where not only the underlying type but also its successor-map is independent of $\mathbb{N}$ (we come to the general form of recursion further below).

So consider any F-algebra $\big(D, (0_D, \mathrm{succ}_D)\big) \in F\,\mathrm{Alg}(\mathcal{C})$ (2), hence an object $D \in \mathcal{C}$ equipped with a morphism

$$0_D : * \to D$$

and a morphism

$$\mathrm{succ}_D : D \to D\,.$$

By initiality of the F-algebra $\mathbb{N}$, there is then a (unique) morphism

$$\mathrm{rec} : \mathbb{N} \to D$$

such that the following diagram commutes:

$$\begin{array}{ccccc} * & \overset{0}{\longrightarrow} & \mathbb{N} & \overset{\mathrm{succ}}{\longrightarrow} & \mathbb{N} \\ \big\downarrow && \big\downarrow{}^{\mathrm{rec}} && \big\downarrow{}^{\mathrm{rec}} \\ * & \underset{0_D}{\longrightarrow} & D & \underset{\mathrm{succ}_D}{\longrightarrow} & D \end{array}$$

This means precisely that rec is the function defined recursively by

$$\mathrm{rec}(0) \;=\; 0_D$$

and

$$\mathrm{rec}\big(\mathrm{succ}(n)\big) \;=\; \mathrm{succ}_D\big(\mathrm{rec}(n)\big)\;. \tag{3}$$

More generally, consider an F-algebra in the slice over $\big(\mathbb{N}, (0, \mathrm{succ})\big)$, but with the underlying slice object assumed (dropping also this assumption leads to the fully general notion of induction further below) to be independent of $\mathbb{N}$, hence of the form

$$\left[\begin{array}{c} \mathbb{N} \times D \\ \big\downarrow{}^{\mathrm{pr}_{\mathbb{N}}} \\ \mathbb{N} \end{array} \right] \;\in\; \mathcal{C}_{/\mathbb{N}}\,, \tag{4}$$

while the F-algebra structure may now depend on $\mathbb{N}$:

$$\begin{array}{ccc} * \sqcup (\mathbb{N} \times D) & \xrightarrow{\big((0, \mathrm{succ} \circ pr_{\mathbb{N}}),\, (0_D, \mathrm{succ}_D)\big)} & \mathbb{N} \times D \\ \big\downarrow{}^{* \sqcup \mathrm{pr}_{\mathbb{N}}} && \big\downarrow{}^{\mathrm{pr}_{\mathbb{N}}} \\ * \sqcup \mathbb{N} & \underset{(0, \mathrm{succ})}{\longrightarrow} & \mathbb{N} \end{array}$$

in that

$$\mathrm{succ}_D \;:\; \mathbb{N} \times D \to D$$

may depend on $\mathbb{N}$.

Now, since with $\big(\mathbb{N}, (0, \mathrm{succ})\big)$ being the initial object in $F\,\mathrm{Alg}(\mathcal{C})$, the identity morphism on $\big(\mathbb{N}, (0, \mathrm{succ})\big)$ is the initial object in the slice category $F\,\mathrm{Alg}(\mathcal{C})_{/(\mathbb{N}, (0, \mathrm{succ}))}$ (cf. there), it follows that from such data is induced a unique morphism f in the following commuting diagram:

$$
\begin{tikzcd}
 & & \ast \sqcup \mathbb{N} \arrow[r, dashed, "\mathrm{id}_\ast \sqcup (\mathrm{id}_{\mathbb{N}}, \mathrm{rec})"] \arrow[dr, equal] \arrow[ddll, "{(0,\mathrm{succ})}"'] & \ast \sqcup (\mathbb{N} \times D) \arrow[d, "\mathrm{id}_\ast \sqcup \mathrm{pr}_{\mathbb{N}}"] \arrow[ddll, "{\big((0, \mathrm{succ} \circ \mathrm{pr}_{\mathbb{N}}), (0_D, \mathrm{succ}_D)\big)}"'] \\
 & & & \ast \sqcup \mathbb{N} \arrow[ddll, "{(0,\mathrm{succ})}"] \\
\mathbb{N} \arrow[r, dashed, "{(\mathrm{Id}_{\mathbb{N}}, \mathrm{rec})}"] \arrow[dr, equal] & \mathbb{N} \times D \arrow[d] & & \\
 & \mathbb{N} & &
\end{tikzcd}
$$

Here the commutativity of the top square means equivalently that

$$\mathrm{rec}(0) \;=\; 0_D$$

and

$$\mathrm{rec}\big(\mathrm{succ}(n)\big) \;=\; \mathrm{succ}_D\big(n, \mathrm{rec}(n)\big)\;. \qquad (5)$$

Remark 3.2. (***the need for dependent recursion*** [cf. Paulin-Mohring (1993, p. 330)])
The appearance of the argument "n" on the right of (5) – in contrast to formula (3) for non-dependent recursion – means (in view of the argument "$\mathrm{succ}(n)$" on the left) that the recursor succ_D has access to the *predecessor* partial function $\mathrm{pred} : \mathrm{succ}(n) \;\mapsto\; n$. This is necessary in order to express all computable functions on the natural numbers inductively and hence explains the need for the dependently typed recursion principle (4)

Induction

Dropping the above constraint (4) on the dependent F-algebra, we spell out in detail how the fact that $\mathbb{N}$ satisfied Def. 2.1 is the classical induction principle.

That principle says informally that if a proposition P depending on the natural numbers is true at $n = 0$ and such that if it is true for some n then it is true for $n + 1$, then it is true for all natural numbers.

Here is how this is formalized in type theory and then interpreted in some suitable ambient category $\mathcal{C}$.

First of all, that P is a proposition depending on the natural numbers means that it is a dependent type

$$n : \mathbb{N} \;\;\vdash\;\; P(n) : \mathrm{Type}\;.$$

The categorical interpretation of this is by a display map

$$\begin{tikzcd} P \arrow[d, "\pi_P"] \\ \mathbb{N} \end{tikzcd} \tag{6}$$

in the given category $\mathcal{C}$.

With this, the commuting diagram which interprets the induction principle is the following:

$$\begin{tikzcd}
 & & \ast \sqcup \mathbb{N} \arrow[r, dashed, "\mathrm{id}_\ast \sqcup \mathrm{ind}"] \arrow[ddll, "(0,\mathrm{succ})"'] \arrow[dr, equal] & \ast \sqcup P \arrow[d, "\mathrm{id}_\ast \sqcup \pi_P"] \arrow[ddll, "(0_P, \mathrm{succ}_P)"'] \\
 & & & \ast \sqcup \mathbb{N} \arrow[ddll, "(0,\mathrm{succ})"] \\
\mathbb{N} \arrow[r, dashed, "\mathrm{ind}"] \arrow[dr, equal] & P \arrow[d, "\pi_P"] & & \\
 & \mathbb{N} & &
\end{tikzcd}$$

We now unwind again how this comes about and what it all means:

First, the fact that P holds at 0 means that there is a (proof-)term

$$\vdash \; 0_P : P(0) \; .$$

In the categorical semantics the substitution of n for 0 that gives $P(0)$ is given by the pullback of the given display map (6):

$$\begin{tikzcd}
0^* P \arrow[r] \arrow[d] & P \arrow[d] \\
\ast \arrow[r, "0"'] & \mathbb{N}
\end{tikzcd}$$

and the term 0_P is interpreted as a section of the resulting fibration over the terminal object

$$\begin{tikzcd}
\ast \arrow[r, "p_0"] \arrow[dr] & 0^* P \arrow[r] \arrow[d] & P \arrow[d] \\
 & \ast \arrow[r, "0"'] & \mathbb{N}
\end{tikzcd} .$$

But by the defining universal property of the pullback, this is equivalently just a commuting diagram

$$\begin{array}{ccc} * & \overset{p_0}{\longrightarrow} & P \\ \downarrow && \downarrow \\ * & \underset{0}{\longrightarrow} & \mathbb{N} \end{array} \,.$$

Next the induction step. Formally it says that for all $n \in \mathbb{N}$ there is an implication $\mathrm{succ}_P(n) : P(n) \to P(n+1)$

$$n \in \mathbb{N} \quad \vdash \quad \mathrm{succ}_P(n) : P(n) \to P(n+1) \;.$$

The categorical semantics of the substitution of $n+1$ for n is given by the pullback

$$P\big(* \sqcup (-) \big) \coloneqq \quad \begin{array}{ccc} s^* P & \longrightarrow & P \\ \downarrow && \downarrow \\ \mathbb{N} & \underset{s}{\longrightarrow} & \mathbb{N} \end{array}$$

and the interpretation of the implication term $\mathrm{succ}_P(n)$ is as a morphism $P \to s^* P$ in $\mathcal{C}_{/\mathbb{N}}$

$$\begin{array}{ccccc} P & \overset{p_s}{\longrightarrow} & s^* P & \longrightarrow & P \\ & \searrow & \downarrow && \downarrow \\ && \mathbb{N} & \overset{s}{\longrightarrow} & \mathbb{N} \end{array} \,.$$

Again by the universal property of the pullback this is equivalently a commuting diagram

$$\begin{array}{ccc} P & \overset{\mathrm{succ}_P}{\longrightarrow} & P \\ \downarrow && \downarrow \\ \mathbb{N} & \underset{s}{\longrightarrow} & \mathbb{N} \end{array} \,.$$

In summary this shows that

- P being a proposition depending on natural numbers which holds at 0 and which holds at $n+1$ if it holds at n

is interpreted precisely as an endofunctor-algebra homomorphism

$$\begin{array}{c} P \\ \downarrow \\ \mathbb{N} \end{array} .$$

for the endofunctor F (1).

The induction principle is supposed to deduce from this that P holds for every n, hence that there is a proof $\mathrm{ind}(n) : P(n)$ for all n:

$$n : \mathbb{N} \quad \vdash \quad \mathrm{ind}(n) : P(n) \;.$$

The categorical interpretation of this is as a morphism $p : \mathbb{N} \to P$ in $\mathcal{C}_{/\mathbb{N}}$. The existence of this is indeed exactly what the interpretation of the elimination rule (Def. 2.1) gives exactly what the initiality of the F-algebra $\mathbb{N}$ gives.

4. Related concepts

- natural numbers, natural numbers object
- decimal numeral representation of the natural numbers
- dependent sequence type
- type of finite types

5. References

Original articles with emphasis on the nature of $\mathbb{N}$ as an inductive type:

- Per Martin-Löf (notes by Giovanni Sambin), pp. 38 of: *Intuitionistic type theory*, Lecture notes Padua 1984, Bibliopolis, Napoli (1984) [pdf, pdf]
- Thierry Coquand, Christine Paulin, p. 52-53 in: *Inductively defined types*, COLOG-88 Lecture Notes in Computer Science ***417***, Springer (1990) 50-66 [doi:10.1007/3-540-52335-9_47]
- Christine Paulin-Mohring, §1.3 in: *Inductive definitions in the system Coq – Rules and Properties*, in: *Typed Lambda Calculi and Applications* TLCA 1993, Lecture Notes in Computer Science ***664*** Springer (1993) [doi:10.1007/BFb0037116]
- Peter Dybjer, §3 in: *Inductive families*, Formal Aspects of Computing ***6*** (1994) 440–465 [doi:10.1007/BF01211308, doi:10.1007/BF01211308, pdf]

The syntax in Coq:

- Christine Paulin-Mohring, p. 6 in: *Introduction to the Calculus of Inductive Constructions*, contribution to: *Vienna Summer of Logic* (2014) [hal:01094195, pdf, pdf slides]

See also:

- Frank Pfenning, *Lecture notes on natural numbers* (2009) [pdf, pdf]

Discussion in a context of homotopy type theory and in view of higher inductive types:

- Univalent Foundations Project, §1.9 in: *Homotopy Type Theory – Univalent Foundations of Mathematics* (2013) [web, pdf]
- Egbert Rijke, §3 in: *Introduction to Homotopy Type Theory*, Cambridge Studies in Advanced Mathematics, Cambridge University Press (arXiv:2212.11082)
- Kajetan Söhnen, §2.4.5 in: *Higher Inductive Types in Homotopy Type Theory*, Munich (2018) [pdf, pdf]
- Peter LeFanu Lumsdaine, Mike Shulman, *Semantics of higher inductive types*, Math. Proc. Camb. Phil. Soc. ***169*** (2020) 159-208 [arXiv:1705.07088, talk slides pdf, doi:10.1017/S030500411900015X]

Equivalence to binary presentations:

- Nicolas Magaud?, Yves Bertot?, *Changing Data Structures in Type Theory: A Study of Natural Numbers*, in *Types for Proofs and Programs. TYPES 2000*, Lecture Notes in Computer Science ***2277*** [doi:10.1007/3-540-45842-5_12, pdf]
- Nicolas Magaud?, *Changing Data Representation within the Coq*, in *Theorem Proving in Higher Order Logics. TPHOLs 2003*, Lecture Notes in Computer Science ***2758*** [doi:10.1007/10930755_6]

That one can construct the natural numbers type from the integers type can be found in:

- Christian Sattler, *Natural numbers from integers* (pdf)

Last revised on February 4, 2024 at 22:08:41. See the history of this page for a list of all contributions to it.

Edit Discuss Previous revision Changes from previous revision History (28 revisions) Cite Print Source