Constructive Notions of Ordinals in Homotopy Type Theory*

Nicolai Kraus[1], Fredrik Nordvall Forsberg[2], and Chuangjie Xu[3]

[1] University of Nottingham, Nottingham, UK
[2] University of Strathclyde, Glasgow, UK
[3] fortiss GmbH, Munich, Germany

Introduction Ordinals are numbers that, although possibly infinite, share an important property with the natural numbers: every decreasing sequence necessarily terminates. This makes them a powerful tool when proving that processes terminate, or justifying induction and recursion [DM79, Flo67]. There is also a rich theory of arithmetic on ordinals, generalising the usual theory of arithmetic on the natural numbers. Unfortunately, the standard definition of ordinals is not very well-behaved constructively, and the notion fragments into a number of inequivalent definitions, each with pros and cons. We consider three different constructive notions in homotopy type theory, and show how they relate to each other.

Cantor Normal Forms as a Subset of Binary Trees In classical set theory, it is well known that every ordinal α can be written uniquely in Cantor normal form

$$\alpha = \omega^{\beta_1} + \omega^{\beta_2} + \cdots + \omega^{\beta_n} \text{ with } \beta_1 \geq \beta_2 \geq \cdots \geq \beta_n \tag{1}$$

for some natural number n and ordinals β_i. If $\alpha < \varepsilon_0$, then $\beta_i < \alpha$, and we can represent α as a finite binary tree (with a condition) as follows [Buc91, NXG20]. Let $\mathcal{T}$ be the type of unlabeled binary trees, i.e. the inductive type with suggestively named constructors $0 : \mathcal{T}$ and $\omega^{-} + - : \mathcal{T} \times \mathcal{T} \to \mathcal{T}$. Let the relation $<$ be the lexicographical order, i.e. generated by the following clauses:

$$0 < \omega^a + b \qquad a < c \to \omega^a + b < \omega^c + d \qquad b < d \to \omega^a + b < \omega^a + d.$$

We have the map $\mathsf{left} : \mathcal{T} \to \mathcal{T}$ defined by $\mathsf{left}(0) :\equiv 0$ and $\mathsf{left}(\omega^a + b) :\equiv a$ which gives us the left subtree (if it exists) of a tree. A tree is a *Cantor normal form* (CNF) if, for every $\omega^s + t$ that the tree contains, we have $\mathsf{left}(t) \leq s$, where $s \leq t :\equiv (s < t) \uplus (s = t)$; this enforces the condition in (1). Formally, the predicate isCNF is defined inductively by the two clauses

$$\mathsf{isCNF}(0) \qquad \mathsf{isCNF}(s) \to \mathsf{isCNF}(t) \to \mathsf{left}(t) \leq s \to \mathsf{isCNF}(\omega^s + t).$$

We write $\mathsf{Cnf} :\equiv \Sigma(t : \mathcal{T}).\mathsf{isCNF}(t)$ for the type of Cantor normal forms.

Brouwer Trees as a Quotient Inductive-Inductive Type In the functional programming community, it is popular to consider *Brouwer ordinal trees* $\mathcal{O}$ as inductively generated by zero, successor and a "supremum" constructor $\mathsf{sup} : (\mathbb{N} \to \mathcal{O}) \to \mathcal{O}$ which forms a new tree for every countable sequence of trees [Bro26, CHS97, Han00]. By the inductive nature of the definition, constructions on trees can be carried out by giving one case for zero, one for successors, and one for suprema, just as in the classical theorem of transfinite induction. However, calling the constructor sup is wishful thinking; $\mathsf{sup}(s)$ does not faithfully represent the suprema of

*Supported by the Royal Society, grant reference URF\R1\191055, the UK National Physical Laboratory Measurement Fellowship project *Dependent types for trustworthy tools*, and the LMUexcellent program.

the sequence s, since we do not have that e.g. $\mathsf{sup}(s_0, s_1, s_2, \ldots) = \mathsf{sup}(s_1, s_0, s_2, \ldots)$ — each sequence gives rise to a new tree, rather than identifying trees representing the same supremum.

Using a *quotient inductive-inductive type* [ACD+18], we can remedy the situation: Let A be a type and $\prec : A \to A \to \mathsf{hProp}$. For sequences $f, g : \mathbb{N} \to A$, we say that f is *simulated by* g if $f \precsim g :\equiv \forall k.\exists n.f(k) \prec g(n)$ (where $\exists$ is truncated Σ). We say that f and g are *bisimilar* with respect to $\prec$, written $f \approx^{\prec} g$, if we have both $f \precsim g$ and $g \precsim f$. A sequence $f : \mathbb{N} \to A$ is *increasing* with respect to $\prec$ if we have $\forall k.f(k) \prec f(k+1)$. We write $\mathbb{N} \xrightarrow{\prec} A$ for the type of $\prec$-increasing sequences. We now mutually construct the type $\mathsf{Brw} : \mathsf{hSet}$ together with a relation $\leq : \mathsf{Brw} \to \mathsf{Brw} \to \mathsf{hProp}$. The constructors for Brw are $\mathsf{zero} : \mathsf{Brw}$, $\mathsf{succ} : \mathsf{Brw} \to \mathsf{Brw}$, and

$$\mathsf{limit} : (\mathbb{N} \xrightarrow{\leq} \mathsf{Brw}) \to \mathsf{Brw} \qquad \text{and} \qquad \mathsf{bisim} : f \approx^{\leq} g \to \mathsf{limit}\, f = \mathsf{limit}\, g,$$

where we denote $x < y :\equiv \mathsf{succ}\, x \leq y$ in the type of limit. The constructors for $\leq$ ensure transitivity, that zero is minimal, that succ is monotone, and that $\mathsf{limit}\, f$ is the least upper bound of f. Because of the infinitary constructor limit, we lose full decidability of equality and order relations, but by restricting to limits of increasing sequences, we retain the possibility of classifying an ordinal as zero, a successor, or a limit.

Extensional Wellfounded Orders Finally, we consider a variation on the classical set-theoretical axioms for ordinals more suitable for a constructive treatment [Tay96], following the HoTT book [Uni13, Chapter 10] and Escardó [Esc21]. The type Ord consists of a type X together with a relation $\prec : X \to X \to \mathsf{hProp}$ which is *transitive*, *extensional* (any two elements of with the same predecessors are equal), and *wellfounded* (every element is accessible, where accessibility is the least relation such that x is accessible if every $y \prec x$ is accessible.).

We also have a relation on Ord itself. Following [Uni13, Def 10.3.11 and Cor 10.3.13], a *simulation* between ordinals $(X, \prec_X)$ and $(Y, \prec_Y)$ is a monotone function $f : X \to Y$ such that for all $x : X$ and $y : Y$, if $y \prec_Y f\, x$, then we have an $x_0 \prec_X x$ such that $f\, x_0 = y$. We write $X \leq Y$ for the type of simulations between $(X, \prec_X)$ and $(Y, \prec_Y)$. Given an ordinal $(X, \prec)$ and $x : X$, the *initial segment* of elements below x is given as $X_{/x} :\equiv \Sigma(y : X).y \prec x$. A simulation $f : X \leq Y$ is *bounded* if we have $y : Y$ such that f induces an equivalence $X \simeq Y_{/y}$. We write $X < Y$ for the type of bounded simulations.

Results For each of Cnf, Brw, Ord, the relation $<$ is transitive, extensional, and wellfounded; for wellfoundedness, the refined definitions of Cnf and Brw which excludes "junk" terms are crucial. For Cnf, $<$ is decidable, whereas for Ord, $<$ is decidable if and only if the law of excluded middle holds. Brw sit in the middle, with some of its properties being decidable, e.g. it is decidable whether a given x is finite, but $<$ is not decidable in general without further assumptions. We introduce an abstract framework axiomatising properties such as being a successor or a limit ordinal, which makes it possible to compare the different notions of ordinals above. According to these definitions, each of Cnf, Brw, Ord has zeroes and successors, and the successor functions of Cnf and Brw are both $<$- and $\leq$-monotone. For the successor function of Ord, each of the two monotonicity properties on its own is equivalent to the law of excluded middle. Cnf does not have limits, but both Brw and Ord do. Using the abstract notions of zero, successor and limit, we can give an abstract specification of the arithmetic operations; we say that a notion of ordinals has unique arithmetic if the type of implementations of the specification is contractible. Cnf has addition, multiplication, and exponentiation with base ω (all unique), Brw has addition, multiplication and exponentiation with every base (all unique), and Ord has addition and multiplication. Finally, we have order-preserving embeddings $\mathsf{Cnf} \hookrightarrow \mathsf{Brw} \hookrightarrow \mathsf{Ord}$.

Details and Formalisation Full details: `arxiv:2104.02549`. We have formalised our results in cubical Agda: `https://bitbucket.org/nicolaikraus/constructive-ordinals-in-hott`.

References

[ACD+18] Thorsten Altenkirch, Paolo Capriotti, Gabe Dijkstra, Nicolai Kraus, and Fredrik Nordvall Forsberg. Quotient inductive-inductive types. In Christel Baier and Ugo Dal Lago, editors, *FoSSaCS '18*, pages 293–310. Springer, 2018.

[Bro26] L. E. J. Brouwer. Zur begründung der intuitionistische mathematik III. *Mathematische Annalen*, 96:451–488, 1926.

[Buc91] Wilfried Buchholz. Notation systems for infinitary derivations. *Archive for Mathematical Logic*, 30:227–296, 1991.

[CHS97] Thierry Coquand, Peter Hancock, and Anton Setzer. Ordinals in type theory. Invited talk at Computer Science Logic (CSL), `http://www.cse.chalmers.se/~coquand/ordinal.ps`, 1997.

[DM79] Nachum Dershowitz and Zohar Manna. Proving termination with multiset orderings. *Communications of the ACM*, 22(8):465–476, 1979.

[Esc21] Martín Escardó. Agda implementation: Ordinals, Since 2010–2021. `https://www.cs.bham.ac.uk/~mhe/TypeTopology/Ordinals.html`.

[Flo67] Robert W. Floyd. Assigning meanings to programs. In J.T. Schwartz, editor, *Symposium on Applied Mathematics*, volume 19, pages 19–32, 1967.

[Han00] Peter Hancock. *Ordinals and Interactive Programs*. PhD thesis, University of Edinburgh, 2000.

[NXG20] Fredrik Nordvall Forsberg, Chuangjie Xu, and Neil Ghani. Three equivalent ordinal notation systems in cubical Agda. In *CPP '20*, pages 172–185. ACM, 2020.

[Tay96] Paul Taylor. Intuitionistic sets and ordinals. *Journal of Symbolic Logic*, 61(3):705–744, 1996.

[Uni13] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. `http://homotopytypetheory.org/book/`, Institute for Advanced Study, 2013.