Constructivity Aspects of Brouwer Tree Ordinals

Nicolai Kraus\textsuperscript{1}, Fredrik Nordvall Forsberg\textsuperscript{2}, and Chuangjie Xu\textsuperscript{3}

\textsuperscript{1} University of Nottingham, UK  \textsuperscript{2} University of Strathclyde, UK  \textsuperscript{3} fortiss GmbH, Germany

In a constructive setting, no concrete formulation of ordinals can simultaneously have all the properties that one might be interested in. For example, being able to calculate limits of sequences is constructively incompatible with deciding extensional equality, because they together implies Bishop’s weak limited principle of omniscience (WLPO) [2]. In this talk, we introduce a refined version of Brouwer trees and discuss the properties that they have and cannot have constructively, using homotopy type theory [8] as the foundational setting.

Brouwer trees in functional programming are often inductively generated by the usual constructors of natural numbers (zero and successor) and a supremum constructor \texttt{sup} that forms a new tree for every countable sequence of Brouwer trees [3,4,5]. 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, it is a priori wishful thinking to call the constructor \texttt{sup} a “supremum”; \texttt{sup}(s) does not faithfully represent the suprema of the sequence \(s\), since we do not have that e.g. \(\texttt{sup}(s_0, s_1, s_2, \ldots) = \texttt{sup}(s_1, s_0, s_2, \ldots)\) — each sequence gives rise to a new tree, rather than identifying trees that would be supposed to represent the same supremum.

We fix this shortcoming in homotopy type theory via a quotient inductive-inductive type [1]: We mutually construct a type \(\text{Brw}\) of Brouwer trees together with a relation \(\leq\) on them. The constructors for \(\text{Brw}\) include \texttt{zero} : \(\text{Brw}\), \texttt{succ} : \(\text{Brw} \to \text{Brw}\), and

\[
\text{limit} : (N < \rightarrow \text{Brw}) \to \text{Brw} \quad \text{and} \quad \text{bisim} : f \approx \leq g \to \text{limit } f = \text{limit } g,
\]

where \(N < \rightarrow \text{Brw}\) is the type of increasing sequences with respect to the relation \(<\) defined by \(x < y \equiv \texttt{succ } x \leq y\), and \(f \approx \leq g\) is the type expressing that the sequences \(f\) and \(g\) are bisimilar with respect to the relation \(\leq\). The constructors for \(\leq\) ensure transitivity, that \texttt{zero} is minimal, that \texttt{succ} is monotone, and that \texttt{limit} \(f\) is the least upper bound of \(f\). We also include “truncation” constructors which ensure that \(\text{Brw}\) is a set and that \(x \leq y\) is a proposition.

The path constructor \texttt{bisim} identifies Brouwer trees that represent the same limit. By restricting to limits of strictly increasing sequences, we retain the possibility of classifying an ordinal as zero, a successor, or a limit. Both relations \(<\) and \(\leq\) are extensional and transitive; \(\leq\) is reflexive and antisymmetric; and \(<\) is irreflexive and well-founded. Moreover, the implications \(x < y \leq z \to x < z\) and \(x \leq y < z \to x < z\) both hold (recall that the latter fails in some constructive formulations of ordinals [7]). The standard arithmetic operations on Brouwer
trees can be implemented by recursion on the second argument. But there are several additional difficulties which stem from the fact that our definition of Brouwer trees identifies limits of bisimilar sequences. Perhaps surprisingly, even though Brw has addition with expected properties, Brw has subtraction if and only if the limited principle of omniscience (LPO) holds.

Because we can distinguish constructors, it is decidable if a Brouwer tree $x$ is finite, i.e. $x < \omega$ where $\omega \equiv \text{limit}(\lambda i . i)$. Furthermore most properties of finite Brouwer trees are decidable. However, deciding equality with $\omega$ is equivalent to WLPO, and deciding equality and the relations $<$ and $\leq$ for arbitrary Brouwer trees is equivalent to LPO. Moreover, trichotomy (i.e. $x < y \lor x = y \lor y < x$) and splitting of $\leq$ (i.e. $x \leq y \rightarrow x < y \lor x = y$) are also equivalent to LPO.

More details of the above results can be found in our preprint [6] which is accompanied with a formalization in cubical Agda (https://bitbucket.org/nicolaikraus/constructive-ordinals-in-hott).

References