ORDINALS IN TYPE THEORY

Thierry Coquand, Peter Hancock and Anton Setzer

Invited talk for CSL'97, Aarhus, August 1997

Motivations

We present a possible definition of "ordinal strength" of a formal system, and the interest of such a notion.

We explain a method for building larger and larger ordinals in type theory.

We try to give a concrete illustration of the difference between predicative/impredicative principles.

Strength of a Formal System

One main use of ordinals in proof theory is to measure the **strength** of a formal system.

What does that mean?

A key point (Richard, Skolem antinomy):

Formal systems are *countable*.

Hence the capacity to denote objects in *uncountable* domains is a priori limited.

Two canonical uncountable domains are:

- the domain of countable ordinals O,
- the set of numerical functions N $\to$N.

Hence extremely natural questions are: how much of O and of N $\to$N can I express in a given formal system?

In the case of O this corresponds to the proof theoretical ordinal which measures how "strong" is the formal system.

Ordinal of a formal system

Usually this notion of ordinal measure applies to a *logical* system.

Here we will apply this notion to a system which can also be seen as a purely *functional* system, that is a system representing functions, functionals at various types. This definition applies also to a large class of functional systems, like Gödel system T (primitive recursion at higher types).

One adds an abstract data type O with the basic operations:

- $0 : \mathsf{O}$,
- $S : \mathsf{O} \to \mathsf{O}$,
- $L : [\mathsf{N} \to \mathsf{O}] \to \mathsf{O}$.

Question: what are the ordinals that can be represented by a closed object of type O?

Notice: **no** elimination rules/recursion operators over O.

Example: Gödel System T

The types are

$$\mathsf{N},\ \mathsf{N} \to \mathsf{N},\ [\mathsf{N} \to \mathsf{N}] \to \mathsf{N}, \ldots$$

but also

$$\mathsf{O},\ \mathsf{N} \to \mathsf{O},\ [\mathsf{N} \to \mathsf{O}] \to \mathsf{O}, \ldots$$

We can think of N as the set of natural numbers, and O as the set of all set-theoretical ordinals.

We represent finite ordinals by the terms:

$$0,\ S\ 0,\ S\ \left(S\ 0\right),\ S\ \Big(S\ \left(S\ 0\right)\Big), \ldots$$

By primitive recursion, we can define $f : \mathsf{N} \to \mathsf{O}$

$$\begin{aligned} f\ 0 &= 0 \\ f\ (n+1) &= S\ (f\ n)\,. \end{aligned}$$

and hence we get up to $\omega = L\ f$.

Gödel System T (continued)

Starting from ω we can represent

$$\omega + 1,\ \omega + 2, \ldots$$

and we can define $g : \mathsf{N} \to \mathsf{O}$

$$\begin{array}{rcl} g\ 0 & = & \omega \\ g\ (n+1) & = & S\ (g\ n)\,. \end{array}$$

hence we get a term representing for $\omega + \omega = \omega.2$

Similarly, we represent $\omega.3,\ \omega.4, \ldots$

Gödel System T (continued)

But it is clear how to do this in an uniform way, because we can use **parameters** in a primitive recursive definition. So we define $f : \mathsf{N} \to \mathsf{O} \to \mathsf{O}$

$$\begin{aligned} f\ 0\ x &= x \\ f\ (n+1)\ x &= S\ (f\ n\ x)\,. \end{aligned}$$

This is exactly what is needed in the step $\omega.n \longmapsto \omega.(n+1)$. We can now take

$$\begin{aligned} h\ 0 &= 0 \\ h\ (n+1) &= L\left(\lambda k\ f\ k\ (h\ n)\right) \end{aligned}$$

and we have

$$h\ 0 = 0, \quad h\ 1 = \omega, \quad h\ 2 = \omega.2, \ldots, h\ n = \omega.n, \ldots$$

so that we represent

$$\omega^2 = L\ h.$$

Problem: How to get ω^ω?

But there is still a long way from there to ω^ω.

The problem is the step $\omega^n \longmapsto \omega^{n+1}$.

This illustrates well one main feature of ordinal notation systems: we want to capture *uniformities*, and this becomes more and more complex.

Notice that this is a functional programming problem: we want to find a program of type O representing ω^ω.

This suggests a possible analysis of the problem: cannot we replace the search of this *program* by the search of a *proof*?

A Similar Problem

Problem: Given a predicate A on natural numbers such that

- $A(0)$,
- $A(x) \rightarrow A(x+1)$,

give a short proof of $A(2^7)$.

This proof should not use induction.

Solution:

Archimedes "The Sand Reckoner",

Gentzen's trick, in proof theory.

Define $A'(x)$ to mean $\forall y. A(y) \rightarrow A(y+x)$.

Notice that:

- $A'(x)$ implies $A(x)$,
- $A'(0)$,
- $A'(x)$ implies $A'(x+1)$,
- $A'(x_1),\ A'(x_2)$ imply $A'(x_1 + x_2)$.

Hence if we define $A_1(x)$ to be $A'(2^x)$ we have $A_1(0)$ and $A_1(x)$ implies $A_1(x+1)$. Hence we have successively

$$A_1\,(1)\,,\ A_1\,(2)\,,\ A_1\,(3)\,,\ A_1\,(4)\,,\ A_1\,(5)\,,\ A_1\,(6)$$

and finally $A_1(7)$, which implies $A(2^7)$.

Sand Reckoner for Ordinals

A predicate on ordinals is an *accessibility* predicate iff

- $A(0)$,
- $A(x) \rightarrow A(x+1)$,
- $[\forall n. A(u_n)] \rightarrow A(\mathsf{sup}\ (u_n))$.

Can we prove *without* transfinite induction $A(\omega^\omega)$ holds?

Gentzen's trick.

We define $A'(x)$ to be $A(y) \rightarrow A(y+x)$ and $A_1(x)$ to be $A'(\omega^x)$.

A' is closed by addition, so $A'(x)$ implies $A'(x.2),\ A'(x.3), \ldots$ and hence $A'(x.\omega)$. From this:

- $A_1(0)$,
- $A_1(x) \rightarrow A_1(x+1)$,
- $[\forall n. A_1(u_n)] \rightarrow A_1(\mathsf{sup}\ (u_n))$.

Hence, we have $A_1(1),\ A_1(2),\ A_1(3), \ldots$ and hence $A_1(\omega)$.

Hence, bf without transfinite induction, we have $A'(\omega^\omega)$ and hence $A(\omega^\omega)$.

How to build ω^ω

By modified realisablity, we get the following functional program.

A limit structure (a, f, l) on a type X consists in

- $a : X$,
- $f : X \to X$,
- $l : [\mathsf{N} \to X] \to X$.

For any limit structure, we can define

$$\omega = l \ \left(n \longmapsto f^n \ a\right).$$

In particular, $(\mathsf{O}, 0, S, L)$ is a given limit structure.

This corresponds to the given predicate $A(\_)$.

How to build ω^ω

On $X' = \mathsf{O} \to \mathsf{O}$

$$a' = \lambda x.x,\ f' = \lambda f.S \circ f,\ l' = \lambda u \lambda x.L\ \ (\lambda n.u\ n\ x)\,,$$

realises that A' is an accessibility predicate.

Similarly, $g : X' \to X' \to X'$

$$g\ x\ y = y \circ x$$

represents the proof that $A'(\_)$ is closed by addition.
And $h : X' \to X'$

$$h\ x = l'\ \ (n \longmapsto x^n)$$

realises $A'(x) \to A'(x.\omega)$.

The limit structure corresponding to $A_1(\_)$ is

$$X_1 = \mathsf{O} \to \mathsf{O},\ a_1 = S,\ f_1 = h,\ l_1 = l'.$$

We get a complex functional program in Gödel System T for ω^ω.

Up to ϵ_0

Can we define $\omega^{\omega^{\omega}}$??

Apply to the predicate $A_1(\_)$ the same transformation

$$A\left(\_\right) \longmapsto A_1\left(\_\right)$$

and define a new predicate:

$$A_2\left(x\right) = \forall y. A_1\left(y\right) \rightarrow A_1\left(y + \omega^x\right).$$

We can prove $A_2(\omega)$ and hence $A_1(\omega^\omega)$ and hence $A(\omega^{\omega^{\omega}})$.

In this way we can represent

$$\omega_1 = \omega,\ \omega_2 = \omega^{\omega_1},\ \ \ldots, \omega_{n+1} = \omega^{\omega_n}, \ldots$$

Hence the ordinal of Gödel's system T is at least ϵ_0.

Application

If we start from the particular limit structure

- $X = \mathsf{N} \rightarrow \mathsf{N}$,
- $a = \lambda n.n + 1$,
- $f = \lambda h.\lambda n.h(n + 1)$,
- $l = \lambda F.\lambda n\ F\ n\ n$,

each ordinal α will be instantiated in a function h_α.

This is the *Hardy hierarchy*.

We have just explained how to represent all h_α for $\alpha < \epsilon_0$ in Gödel System T.

One main step: (Archimedes-)Gentzen's trick.

How to go further?

One way is to use **impredicative** principles.

This is actually mentionned in Gentzen's paper.

We take the intersection of *all* accessibility predicates (impredicatively definition).

This is an accessibility predicate.

It is closed by $x \longmapsto \omega^x$ and hence contains ϵ_0.

A predicative extension

Poincaré's suggestion ("La logique de l'infini", 1909).

Starting from one accessibility predicate A, we build a decreasing sequence of accessibility predicate

$$\ldots A_3 \subseteq A_2 \subseteq A_1 \subseteq A$$

such that

$$A_{n+1}(x) \rightarrow A_n(\omega^x)$$

Take A_ω intersection of all A_n

$$A_\omega(n) = \forall n. A_n(x)$$

A_ω is predicatively defined and closed by $x \longmapsto \omega^x$.

A predicative extension

We need a *logical* system in which we can form predicates such as A_ω.

A logical system which gives precisely this possibility is presented in "The Continuum" (Herman Weyl, 1918).

It is called here the "principle of iteration".
"By far the most complex principle of all, and specifically the *mathematical* one."

It expresses that given a base predicate $A(\_)$ and a predicate transformer $F : X(\_) \longmapsto X_1(\_)$ we can consider the relation $A_n(\_)$ such that $A_0(\_) = A(\_)$ and $A_{n+1}(\_) = F(A_n(\_))$.

This is what we need in order to define A_ω.
Hence, we can represent ϵ_0 in Weyl's logical system.

A predicative extension

We need a *functional* system in which we can represent ϵ_0.

A functional system which gives precisely this possibility is due to Per Martin-Löf.

Extension of Gödel system T with transfinite types, by mean of one universe of small types U.

We can define functions $F : \mathsf{N} \to U$ such as

$$
\begin{array}{rcl}
F\ 0 & = & \mathsf{N} \\
F\ (n+1) & = & F\,(n) \to F\,(n)
\end{array}
$$

and defines types such as $X^{(\omega)} = (\Pi n : \mathsf{N}) F\ n$.

We have only the recursion operator over natural numbers.

How to go further

Surprisingly easy to get a predicative proof of

$$A\left(\epsilon_0\right),\ A\left(\epsilon_1\right),\ldots$$

Define $A_{\omega+1}(x)$ to be $A_\omega(\epsilon_x)$.

We can prove that $A_{\omega+1}$ is an accesibility predicate, and by construction

$$A_{\omega+1}\left(x\right)\ \rightarrow A\left(\epsilon_x\right)$$

So, we have $A_{\omega+1}(0),\ A_{\omega+1}(1),\ldots$ and at the limit $A_{\omega+1}(\omega)$ and hence $A(\epsilon_\omega)$.

How to go further

By realisability we get the following notion.

Let $LS(X)$ is the set of all limit structures on X.

A *lense* is a function F on sets, together with two maps

$$LS\,(X)\ \to LS\,(F\ X)$$

$$F\ X \to X$$

This notion has exteremely good closure properties.

For instance, there is a notion of limit structure on lenses.

How far can we go, predicatively?

Hilbert “On the Infinite” (1925) considered quite similar functional systems. He introduced variable types and types such as X^ω.

He stated (lemma 2) that in defining numerical functions, and ordinals we can always eliminate the use of transfinite recursion and replace it by ordinary recursion.

One example in “The foundations of Mathematics” (1927) was the definitions of ϵ-numbers.

Hancock’s conjecture: with one universe, we can define exactly the ordinals less than $\phi_{\epsilon_0}\ 0$.

Conclusion

Concrete and suggestive representation of ordinals in type systems.

Good example of programming with proofs.

We have all the pieces for a satisfactory solution of the "easy" part of Hancock's conjecture: we can represent at least all ordinals less than $\phi_{\epsilon_0}\ 0$ in Martin-Löf type theory with one universe.

Two key steps

- (Archimedes-)Gentzen's trick,
- Poincaré suggestion for replacing an impredicative intersection by a countable one.

A very indirect complete solution of this problem is due to Peter Aczel (1974).