(Ordinal-theoretic) Proof Theory

Peter Hancock
hancock@spamcop.net

Midlands Graduate School: April 14, 2008

WARNING

THESE NOTES ARE INCOMPLETE, NOT WELL PROOF-READ, AND DO NOT COINCIDE EXACTLY WITH MY COURSE SLIDES.
I should like to develop them into something better organised; I'd be very grateful to anyone who emails me[a] criticisms, suggestions, or even typos. Drop me a line too if you want to know when a decent version is available.

[a] hancock@spamcop.net

Abstract

This course provides an introduction to ordinal-based proof theory, and the notion of proof-theoretic strength particularly for type-theories, or systems which can be seen as type-theories through the Curry-Howard correspondence. A surprisingly large number of number of people express some curiosity about the subject, and this is my excuse for offering it.

The technical content will be divided into 3 pieces,

- the countable ordinals, and their arithmetic including Veblen's hierarchy.
- lower bounds: which is the programming problem of writing programs that denote large ordinals, by exploiting the type-structure available – this will focus on a notion of my own, called a lens; Godel's T will be treated thoroughly, and the use of lenses in connection with universes might be sketched.
- upper bounds: the problem of bounding the size of the ordinals that can be built within a type-theory – this will focus on infinitary term-systems, cut-elimination for sequent calculi, and normalisation for a typed lambda calculus whose types are closed under countably infinite conjunction.

I'll stress connections with programming, where possible. I'll try to promote an algebraic approach to this subject.

1 Why this subject?

Proof theory is nowadays a broad subject. If there is any single question or puzzlement that leads one into it, it might be something like this: what (on earth!) is a proof? There is really a whole cloud of questions here. What makes a proof hang together? What is it for? What use is it? What makes it persuasive? What is an inference step? What makes it work? What is a reason? Or rigour? What is reasoning? If you think these questions are silly, or you know answers to them, count yourself lucky. Study some more respectable subject. By all means though, browse around among some of the fascinating and sometimes beautiful mathematics that has been generated by people with some form of this affliction.

According to Saunders Mac Lane [1] (in an article from 1997 complaining about, amongst other things, proof theory), a real proof is not simply a formalised document, but a sequence of ideas and insights: proof is the very stuff of mathematics. He adds:

> The subject of proof theory should be the understanding and the organization of the various types of insights and their astute combinations which do occur in the construction of mathematical proof. I know of little serious work in this philosophical direction beyond the rather naive attempt in my own Ph.D. thesis (1934), republished in 1979.

Such modesty! What does one do with a proof, thought of like this? You contemplate it, run your mind up and down its rails, until perhaps you behold the theorem, as it were, in a warm friendly light, rather than in baleful cold doubt. This seems to me typical of the mathematician's idea of proof, as a wand for communicating conviction and insight. There is a lot that is right and easily overlooked in this view. A proof is not, or at least not merely a formal object, that could 'in principle' be written out in some syntax and mechanically checked. The distinction is that between understanding and copying a mathematical proof. Still less is it (merely) a peculiar variety of complicated finite combinatorial structure, to be studied in some distant ugly branch of graph theory.

In the light of the Curry-Howard correspondence between proofs/propositions and programs/types, and various developments coming largely from proof theory – cut-elimination, normalisation, functional interpretations, etc. – we now know that there is an intrinsic connection between reasoning and computations. The questions of what a proof is, or what it for, is now a hard question with some point, one we can struggle with.

There are many excellent texts on general proof theory, particularly structural and substructural proof theory, and the subject is vibrant on both sides of the interface between mathematical logic and computing science. I doubt that I could do better than recommend some of these texts for your study.

In these notes I am going to focus on a part of proof theory which has *not* enjoyed such a wide appreciation, namely ordinal-theoretic, or 'ordinally informative' proof theory. The reason for this focus is mainly because I am often asked about it, and pleasurably surprised by the interest and curiosity in it shown by my colleagues. It is also (in part) because I have a particular point of view about it, or drum to bang. The subject is in my opinion in bad need of a thorough algebraic and conceptual overhaul, if only to explain itself better to the educated public, referees of proposals for funding, and not least people interested in the mathematical foundations of computation. You!

I don't want to imply that the proof theorists are doing a particularly bad job of explaining themselves. Some of them (particularly Michael Rathjen and Jeremy Avigad) do a very good job of explaining themselves, and promoting their subject, at least to mathematicians. I don't think they do a particularly good job of engaging with computer scientists. This is a great shame, or at least a disappointment to me. So I thought I'd have a go.

Mathematicians are often happy with a definition or a construction that is repulsive to a programmer: because it is too complex or brutal, or full of ugly special cases, or makes little computational sense. Strangely, this seems to be particularly the case in ordinal theoretic proof theory. 'Strangely' because the subject is very much motivated by computational ideas. Yet it is riddled with nonconstructive arguments, notions and methods. Even where, in a purely mathematical sense it is 'evident' that some argument could 'in principle' be made constructive, the idea of actually teasing out the computational content into a program is often unpleasant even to contemplate. Perhaps the most bitter irony is that the very notion of ordinal, perhaps the most basic notion in the whole subject, is a thoroughly set-theoretical idea; it seems to be a rather deep problem to give a completely satisfactory constructive treatment of the notion of ordinal. [1]

I have found that a valuable aid to really getting a grip on a subject, to formulating its concepts in a smooth and simple way is to try, at least in one's head, to fully formalise the arguments, so that they can be checked on a machine, in some language such as Agda. If you choose the wrong definitions, and you (are fortunate enough to) have a normal human allocation of stamina and brainpower, you will never finish. The thing will explode into thousands of lines of sprawling code. The pressure of formalisation forces you to go back and rethink the most basic definitions, and to simplify and factorise the basic concepts. Of course, there's no certainty you will succeed. I have tried to carry out formalisations in certain parts of proof theory, with partial success: for example, in the direction of the 'lower-bounds' methods mentioned in the abstract, that are essentially a matter of programming. In other directions, things have, so far, not gone so well. There is a challenge here for those of you interested in machine formalisation, particularly in programming languages based on type-theory.

[1] Not many people have tried. An honourable exception is Paul Taylor, who has some interesting papers on the subject. There is little overlap between his ideas and mine. It should also be mentioned that a *recursive* notion of ordinal has been extensively developed by logicians, starting from papers of Church and Kleene in 1938.

1.1 Genesis: Hilbert's Program

The term 'proof theory', in German *beweistheorie*, was (as far as I know) introduced by Hilbert (a giant among mathematicians, like Mac Lane), who in 1925 [2] in a paper whose English title is 'On the Infinite' referred to it thus:

> "Mathematics in a certain sense develops into a tribunal of arbitration, a supreme court that will decide questions of principle – and on such a concrete basis that universal agreement must be obtainable and all assertions can be verified.
>
> Even the assertions of the recent doctrine called 'intuitionism', modest though they may be, can in my opinion obtain their certificate of justification only from this tribunal.
>
> As an example of the way in which fundamental questions can be treated I would like to choose the thesis that every mathematical problem can be solved. We are all convinced of that. After all, one of the things that attracts us most when we apply ourselves to a mathematical problem is precisely that within us we always hear the call: here is the problem, search for the solution ; you can find it by pure thought, for in mathematics there is no *ignoramibus*. Now to be sure, my proof theory cannot specify a general method for solving every mathematical problem ; that does not exist. But the demonstration that the assumption of the solvability of every mathematical problem is consistent falls entirely within the scope of our theory.
>
> I would still like to play a last trump. . . . "

. . . and he goes on to sketch a 'proof' of the continuum hypothesis, that the real numbers can be enumerated by means of the numbers of Cantor's second number class! Silly old Hilbert. (The ideas in this 'proof' are indeed interesting. I'm certain that Gödel's proof of the consistency hypothesis was stimulated by them. We will see another re-use of them later on in due course.)

Note the words 'my proof theory', and the later 'our theory', that presumably refers to the same thing. What was this?

1.1.1 Hilbert's proof theory, and program

The whole thing was based on *worry*, not to say *paranoia*. It was a search for certainty, reliability. The end of the 19th century and the first two or three decades of the 20th was a period of intense ferment[2] among some of the most distinguished mathematicians of the day. Mathematics was afflicted with paradoxes: Frege's, Russell's, ... New methods of proof and new concepts were being introduced – Dedekind's ideals in algebraic number theory, indirect proofs as in Hilbert's own basis theorem, Zermelo's proof that the reals can be well-ordered,

[2] Not only in mathematics.

Cantor's theory of transfinite numbers, the abstract notion of function. Controversy (literally) raged on all sides: Brouwer rejected use of the excluded middle, Poincare and Weyl rejected use of impredicative definitions, and so on and so forth. In Hilbert's words:

> Just think: in mathematics, this paragon of reliability and truth, the very notions and inferences that everyone learns, teaches and uses lead to absurdities. And where else would reliability and truth be found if even mathematical thinking fails?

The aim of Hilbert's proof theory, (or rather his *program(me?)*) was to "endow mathematical method with ... definitive reliability". Hilbert wanted to *save*, or put on solid ground, the use of 'ideal' entities, or infinitistic methods in modern mathematics. How? Where would that certainty or reassurance come from?

For various reasons, going back to the philosopher Kant, Hilbert thought that there was a part of mathematics that "neither can be reduced to anything else nor requires reduction". He called this 'contentual', or 'finitist' mathematics. To cut a long story sort, it is expressed in free variable (implicitly, universally quantified) propositions, in which the variables range over 'concrete', 'surveyable' finite objects, such that each instance can be verified (or falsified) by computation. "This is the basic philosophical position that I consider requisite for mathematics, and in general all scientific thinking, understanding, and communication." Examples of concrete objects were natural numbers, or indeed formal derivations, in a formal system. Examples of contentual propositions are:

- Every even number greater than 2 can be expressed as the sum of a pair of primes.
- For no triple a, b, c of non-zero natural numbers and $n > 2$ is it the case that $a^n + b^n = c^n$.
- The four colour theorem.
- The consistency of propositional calculus. Or Girard's system F. Or ZFC.

Hilbert's Program (for the 'salvation' of mathematics) amounted to this:

- Codify all mathematical reasoning in some formal system T.
- Prove, by finitistic means (F), if some 'real' statement A (like $0 = 1$) is provable using ideal notions or infinitistic methods, then it should be provable by finitistic means (F). A 'conservativity' property:

  $$T \vdash A \implies F \vdash A$$

  It is clearly *necessary* that there should be a finitistic proof of the consistency of T. For reasons not entirely clear to me (and not me alone), it is generally agreed that such a consistency proof would also be *sufficient.* Apparently:

  $$T \vdash A \implies F + \mathit{Con}(T) \vdash A$$

  But I don't see it. (Do you? It is apparently a theorem of Kreisel's, but I have not been able to find a proof.)

1.1.2 Its failure

The destruction of Hilbert's program (which had its heyday in the 1920's) was accomplished by Gödel's celebrated incompleteness theorems, which announced in a lecture in 1930. By the way, Gödel's results would have been unthinkable without Hilbert's metamathematics: the treatment of formal systems, with their formulas and derivations as concrete mathematical objects. (Formal proofs as mathematical objects in their own right.)

Gödel's 1st incompleteness theorem showed that T contains enough of arithmetic, there is a true (real) finitistic A such that

- If T is consistent, then $T \nvdash A$.
- If T satisfies a further condition (ω-consistency), then $T \vdash \neg A$.

So much for the first part of Hilbert's program (a complete axiomatisation of 'all' mathematics). But perhaps that is not fatal? (What do you think?)

Gödel's 2nd incompleteness theorem showed that (moreover):

$$T \nvdash \mathit{Con}(T)$$

This is generally held to be fatal for the second part of Hilbert's program (that a proof of a 'real' statement proceeding via infinitary statements and notions can be transformed into a direct proof, entirely with in F). This seems fairly conclusive, particularly if one expects that finitistic reasoning should be codifible in first order arithmetic. What do you think?

Despite the failure of Hilbert's program, a *modified*, or *extended* version of it has lived on, and enjoyed a fair amount of success. Very roughly, the modification consists in replacing *finitistic* by *constructive*. It is not uncontroversial, but finitism is generally held to be embodied in some variant of primitive recursive arithmetic PRA. Whereas (again, not uncontroversially) constructivity is generally held to be embodied in some version of Martin-Löf's type theory – extended perhaps by strong forms of universe principles. Quite large fragments of classical second-order arithmetic (PA2, and related fragments of set-theory) have been reduced to such a basis. A lot of insight into infinitistic mathematical principles has been obtained by gauging them in terms of such universe principles – or indeed the structure of ordinal representation systems required for their analysis.

1.1.3 Gentzen

In 1936 Gentzen published a (revised version of his first) consistency proof for classical first order arithmetic (PA = Peano Arithmetic). The (later[3]) proof used notations for ordinals in an essential way (as indeed had certain earlier, unsuccessful attempted consistency proofs by Ackermann (one of Hilbert's assistants). Gentzen was himself since 1934 Hilbert's assistant in Göttingen.

[3] Some of you may be interested to know that the first proof – considered dubious by the referees – used ideas with a game theoretical flavour.

Gentzen used transfinite induction, up to the ordinal

$$\epsilon_0 = \sup \{\omega, \omega^\omega, \omega^{\omega^\omega}, \ldots\} = \text{least } \alpha \text{ s.t. } \alpha = \omega^\alpha$$

Since the characteristic axiom scheme of first order arithmetic is numerical induction

$$F(0) \wedge (\forall x)[F(x) \rightarrow F(S(x))] \rightarrow (\forall x)F(x)$$

i.e.,, induction up to ω, smart-alecs among the mathematicians quipped that Gentzen was the man who proved the consistency of induction up to ω by using induction up to ϵ_0. The crucial point though is that in the induction principle of first order arithmetic, the formula $F(x)$ (with free variable x) may be of arbitrary logical complexity – it may contain arbitrarily nested implications and quantifiers. In contrast, Gentzen's proof used transfinite induction up to ϵ_0 only for decidable (in fact, primitive recursive) F – $\mathrm{TI}_{\mathrm{PR}}(\epsilon_0)$; besides this, the proof was entirely 'finitistic' in Hilbert's sense – in fact, could be carried out in primitive recursive arithmetic.

Moreover, slightly later (in 1942) Gentzen (in his 'Habilitation') showed that his consistency result was best possible, in the sense that PA proves transfinite induction (for arbitrary F) for any $\alpha < \epsilon_0$.

The picture to which this gives rise is that the non-finitist ('ideal') part of first order numerical induction is in some sense encapsulated in transfinite induction with respect to primitive recursive predicates, up to (but not including) ϵ_0, and therefore 'measured', or 'gauged' by ϵ_0. A spectacular reduction of logic to (transfinite) arithmetic.

One might therefore be tempted to adopt the following as a definition of the proof-theoretic strength, or consistency strength of a theory T.

$$|T|_{Con} = \text{least } \alpha \text{ s.t. } \mathrm{PRA} + \mathrm{TI}_{\mathrm{PR}}(\alpha) \vdash \mathit{Con}(T)$$

One should be wary of that temptation. In fact matters are extremely subtle: what is important is not so much the size of the ordinal (how large you make it) but its algebraic structure (how you make it large). In fact, you can (if you enjoy that kind of thing) define artificial primitive recursive total orderings on the natural numbers, of order-type ω, such that you can prove the consistency of PA by induction over that ordering. (See e.g., section 10.5 of [3] – such examples, most of which are due to Kreisel, are sometimes referred to as 'dreary pathologies'. See also the early pages of the highly recommended paper [4].)

1.1.4 Resources

There's a particularly good article by Richard Zach in the Stanford Encyclopedia of Philosophy: `http://plato.stanford.edu/entries/hilbert-program/`. It has an extensive bibliography.

There's a fine chapter (24) on Gödel's theorem and its impact on Hilbert's program in the book [5].

The philosopher Panu Raatikainen has a recent paper [6] on the impact of Gödel's theorems at `http://www.mv.helsinki.fi/home/praatika/Hilbert'` s%20Program%20Revisited.pdf. It seems to tease apart the issues rather carefully. There is even a suggestion that it was Brouwer's attack on the excluded middle (rather than, say, strong forms of the comprehension axiom, or impredicative set-theoretical principles) against which Hilbert specifically wanted to secure classical infinitistic methods. This seems interesting in the light of the subsequent successes of the relativised or extended Hilbert's program.

1.2 The infinite

Call a set (type, collection, totality, multiplicity, plurality, ...) A is *countable*, or enumerable, if it can be exhaustively listed $(a_0, a_1, a_2, \ldots)$, where the list may be finite or infinite (in other words, a colist, in the final coalgebra $(\nu X)\, 1 + A \times X)$. (So empty and finite sets come out as countable.) Call any other infinite set *uncountable.*

Two uncountable totalities loom large in the history of mathematics.

- $\mathbb{N}^\mathbb{N}$ – the set of oneplace numerical functions (uncountable by an observation of Cantor). This has the same cardinal as (can be put into bijection with) the continuum, i.e., the set $\mathbb{R}$ of real numbers. (By the way, my habit is to use exponential notation $\mathbb{N}^\mathbb{N}$ and arrow notation $\mathbb{N} \to \mathbb{N}$ interchangeably.)
- Ω – the set of countable ordinals (Cantor's second number class, or to be pedantic, a cumulative version thereof). We will come to this in a moment.

The question of whether $\mathbb{N}^\mathbb{N}$ has the same cardinal Ω is the continuum hypothesis, and is independent of the usual axioms of set-theory (consistent by Gödel 1937, independent by Cohen 1961). We (at least, I) know hardly anything[4] about the cardinal of $Nat^\mathbb{N}$.

A formal system is a countable sort of thing. Its formulas and formal derivations are given by finitary inductive rules (that is, schemas with finitely many premises). A formula, or a proof is a finite, concrete object in Hilbert' sense, and the set of theorems of any particular form is countable. Without getting involved in a lot of detail, we can say that the capacity to denote objects in an uncountable domain is a priori limited. Hence two extremely natural questions that arise for any formal system, albeit not yet sharp, mathematical questions are

- How much of Ω can be expressed?
- How much of $\mathbb{N} \to \mathbb{N}$ can be expressed?

It turns out that these questions match up quite well with two topics in ordinal theoretic proof theory, namely

[4] Only that it is regular. There has been some recent interest among set-theorists – Foreman, Woodin – in settling the cardinal of the continuum on the basis of new 'evident' set-theoretical axioms.

- provable ordinals. (Well orderings that can be proved to be wellfounded in our system.)
- provably recursive functions. (Which Turing machines can be proved to define total recursive functions in our system.)

1.2.1 Ω

If you are a programmer, the most congenial definition of the second number class Ω as a datatype is probably the following.

$$\Omega \triangleq (\mu\, X)\, 1 + X + X^{\mathbb{N}}$$

To a small extent, this is a (white) lie. This is not at all how a set-theorist would describe things.

However these are the things that are usually called the "Brouwer ordinals". A Haskell declaration might look something like this:

```
data Nat   = ZeroN | SuccN Nat
data Omega = ZeroO | SuccO Omega | LimO (N -> Omega)
```

There are at least two reasons for not writing things out in Haskell like this.

- By using the initial-algebra notation $(\mu\, X)\, 1 + X + X^{\mathbb{N}}$, we get to say that we mean the least fixed point (in which all trees are finite-path: all paths from the root to a leaf are finite). By contrast $(\nu\, X)\, 1 + X + X^{\mathbb{N}}$ is a completely different kettle of fish, in which the trees can be 'infinitely deep', as well as 'infinitely wide'. Haskell is based on domain-theoretic ideas, according to which initial and final coalgebras coincide. It is an imperfect medium for expressing mathematical constructions.
- The Haskell notation is cluttered up with names for the constructors. We may have to put up with that if we want to have our definitions processed by a compiler. Few of us (I hope) are so pedantic that we actually want to see the `N`'s and `O`'s that distinguish `ZeroN` from `ZeroO`.

Precisely the same *structure* could be expressed with a different Haskell definition:

```
data Omega = Stop | Wait Omega | Read (N -> Omega)
```

Here the constructor names have been chosen to suggest a computational interpretation of an ordinal. An ordinal is a (data structure which we can interpret as a) *program* of a simple, if rather abstract kind. Either it is terminated (`Stop`), or it `Wait`s for a 'continue' signal before continuing, or it `Read`s a natural number from its environment, and uses the input to choose "where the program counter should go next".

Picture the program counter as starting at the root, and tracing a (finite) path from there to some leaf. As it travels to a leaf, it 'beeps', or hops over a successor constructor to the immediate subtree; and at a `Read` constructor, it uses the input to steer control to one of the countably many subtrees.

Lets see some inhabitants of Ω. First, it is clear how to inject $\mathbb{N}$ into Ω.

```
inj :: Nat -> Omega
inj ZeroN = ZeroO
inj (SuccN k) = SuccO (inj k)
```

Now we can define the first infinite ordinal (or, at any rate a minimal one), namely

```
omega = Lim (\k->inj k)
```

Then we get `SuccO omega`, and so on. We'll come soon to some more powerful machinery for defining greater ordinals.

Before forgetting about Haskell altogether, I want to make one further remark about our third constructor, beit `Lim` or `Read`. I might equally have defined first a notion of stream as a final coalgebra

$$\mathit{Stream}\,A \triangleq (\nu\,X)\,A \times X$$

and then the Brouwer ordinals as

$$\Omega \triangleq (\mu\,X)\,1 + X + \mathit{Stream}\,X$$

It makes no difference.

Now the elements of Ω are clearly all countable, in the sense that each ordinal has countable many structural predecessors. But the type Ω itself is uncountable. For if we have a countable sequence of elements Ω, well, we can form the `Lim` of that sequence, which is a different ordinal (having all the elements of the sequence as immediate predecessors).

The type Ω is sometimes called the second number class. You may guess, and you'd be right, that $\mathbb{N}$ is the *first* number class. This terminology comes from Cantor. You'll probably encounter it. But beware, sometimes people mean slightly different things by this 'number class' talk. Nowadays, most people probably understand number-classes *cumulatively*, so that the second number class contains the first number class. Whereas for Cantor himself, the number classes were disjoint.

To really pin down the number class terminology, I'd have to go into the notion of *cardinal*, which is a side-issue in these notes.

1.2.2 Some more Ω's

Consider now the following type Ω_2.

$$\Omega_2 \triangleq (\mu\,X)\,1 + X + X^{\mathbb{N}} + X^{\Omega}$$

(Imagine there is an invisible subscript $_1$ on our first Ω.)

What are these? Think of them again as abstract programs, with a slightly enlarged repertoire of actions. There are 4 forms

- Terminated.
- Beep. Or, if you prefer, wait for a signal (like an acknowledgement) from the environment, then continue.
- Read a natural number, and use the value read to pick the next node.
- Read a *program* of the first kind, and use it to pick the next node.

Just as we injected $\mathbb{N}$ into Ω, and then formed ω using the injection, plainly we can inject Ω into Ω_2, and form a new ordinal ω_1 that is an element of Ω_2 having each element of $\Omega = \Omega_1$ as an immediate structural predecessor. This ordinal, ω_1 is the first *uncountable* ordinal. Ω_2 is sometimes called the *third* number class.

EXERCISE 1 *Write down definitions for Ω_3 and ω_2.*

That was easy! But the following will probably be a little more taxing.

EXERCISE 2 *Write down definitions for Ω_{113} and ω_{112}. Can you write them in Haskell?*

The issue here is of course not to write it out in detail, which would be insane, but to capture the uniformity of the step from one number class to the next, as an operation on some type. Then express the things you are asked to express using iteration – and perhaps some other ingredients. (Hint: go *up*! I don't myself know the answer as to whether these constructions can be written in Haskell.)

It is natural to call $\mathbb{N}$ the *first* number class, and think of it as Ω_0 – though that is not standard terminology. After all, it is the type preceding the second number class Ω in the sequence

$$\begin{array}{ll} \mathbb{N} & = (\mu\, X)\, 1 + X \\ \Omega & = (\mu\, X)\, 1 + X + (\mathbb{N} \to X) \\ \Omega_2 & = (\mu\, X)\, 1 + X + (\mathbb{N} \to X) + (\Omega \to X) \\ \Omega_3 & = (\mu\, X)\, 1 + X + (\mathbb{N} \to X) + (\Omega_2 \to X) \end{array}$$

If, however, you managed to solve the last exercise, you'll know that it is *not* actually the first – there is another number class before it, which is... the empty type $N_0 = \{=\}\emptyset$! So, in a certain sense $\Omega_{-1} = \emptyset$.

At this point, I hope that (whether or not you have managed to solve the last problem), you have a fairly clear idea how to form all the finite number classes $\mathbb{N}$, Ω, Ω_2, ... and the so-called 'initial' ordinals ω, ω_1, ω_2, ... that are their representatives (inside later number classes). I hope too that you're wondering too whether there are types $Omega_\omega$, ordinals ω_ω, and types and ordinals beyond these. Indeed there are.

You may also be wondering what sort of type systems we need to write these types down. Some of you will probably see how (in principle) to write out the definitions of the finite number classes in system F, or (in practice) in system

F_ω. If anyone can see how to write definitions of Ω_ω in system F or F_ω, (which ought to be possible) please tell me (and/or Thorsten Altenkirch).

However, in the remainder of the course, we will only need the countable ordinals, and won't need the higher number classes (though they are necessary for more advanced topics). Moreover, we will be dealing only with quite 'small' countable ordinals.

2 Arithmetic

In this section, I'm going to use 0, $^+$ and $\sqcup$ are constructors for the countable ordinals. So $\sqcup$ has type $(N \to \Omega) \to \Omega$. To reduce notational noise, I'll take certain liberties in connection with $\sqcup$, treating it as a binding operator, so that I can write $\sqcup_i \ldots i \ldots i \ldots$ instead of something like $\sqcup((\lambda\, i)\ \ldots i \ldots i \ldots)$. Nor will I distinguish (notationally) between the natural numbers and the corresponding finite ordinals in Ω.

Using the initiality properties of Ω, we can define some standard arithmetical functions:

$$
\begin{array}{lcl@{\qquad}lcl}
(+) : \Omega \to \Omega \to \Omega & & & (\times) : \Omega \to \Omega \to \Omega & & \\
\alpha + 0 & \triangleq & \alpha & \alpha \times 0 & \triangleq & 0 \\
\alpha + \beta^+ & \triangleq & (\alpha + \beta)^+ & \alpha \times \beta^+ & \triangleq & (\alpha \times \beta) + \alpha \\
\alpha + \sqcup_i \beta_i & \triangleq & \sqcup_i(\alpha + \beta_i) & \alpha \times \sqcup_i \beta_i & \triangleq & \sqcup_i(\alpha \times \beta_i)
\end{array}
$$

$$
\begin{array}{lcl}
(\uparrow) : \Omega \to \Omega \to \Omega & & \\
\alpha \uparrow 0 & \triangleq & 0^+ \\
\alpha \uparrow \beta^+ & \triangleq & (\alpha \uparrow \beta) \times \alpha \\
\alpha \uparrow \sqcup_i \beta_i & \triangleq & \sqcup_i(\alpha \uparrow \beta_i)
\end{array}
$$

These are the standard arithmetical operations on finite ordinals, but extended to countable ordinals. The extension stipulates that the function commutes with the infinitary operation in its second argument – which is the business end of the function. This commutation property is sometimes called *continuity*.

By the way, I'll assume that each of the operators $+$, $\times$ and $\uparrow$ are right associative, and moreover, that listed here in order of increasing binding strength.

We are now in a position to write down expressions for all ordinals up to the celebrated ordinal ϵ_0.

EXAMPLE 1

$$
\begin{array}{lllll}
0, & 1, & 2, & \ldots & \\
\omega, & \omega + 1, & \omega + 2, & \ldots & \\
\omega \times 2, & \omega \times 2 + 1, & \omega \times 2 + 2, & \ldots & \\
\omega \times 3, & \omega \times 4, & \omega \times 5, & \ldots & \\
\omega \uparrow 2, & \omega \uparrow 2 + \omega + 1, & \ldots & \omega \uparrow 2 + \omega \times 97 + 115, & \ldots \\
\omega \uparrow 3, & \omega \uparrow 4, & \ldots & & \\
\omega \uparrow \omega, & \omega \uparrow (\omega \uparrow \omega), & \ldots & &
\end{array}
$$

2.1 Order relations, and ‘stature’: some problems

However, we aren’t really in a position to state any theorems about ordinals, and so describe the properties of these basic arithmetical operations, because we don’t yet know how to define the order relations $=$, $\leq$, $\geq$, $<$, $>$. Here there is a point of some difficulty. There are in fact *many* possible ways of defining these relations, and it is a rather subtle matter to sort out which definitions to use, at least if we are to stick with the Brouwer ordinals (which are data structures) rather than the set-theoretic ordinals. I will give the definitions that I have settled on (rather recently) in a moment (to show how it can be done), but use a more intuitive pictorially based and thoroughly informal definition that corresponds more closely to the set-theoretic notion.

What is the problem exactly? Brouwer ordinals are data-structures, built up by inductive clauses. Like any other data structure of that kind, there is a relatively clear notion of *structural* predecessor, immediate predecessor, and so on. So for example α is an immediate structural predecessor of α^+, and all of $f(0)$, $f(1)$, $f(2)$, $\ldots$ are immediate structural predecessors of $\sqcup f$. Similarly, there is a relatively clear notion of *extensional* equality between Brouwer ordinals. But these are *not* (or scarcely ever) the notions we want in transfinite arithmetic based on these data-structures.

The point is that ordinals have ‘stature’, or ‘height’ as well as structure. For example, $\sqcup\{0, 2, 4, 6, \ldots\}$ and $\sqcup\{1, 3, 5, 7, \ldots\}$ have the same height. (In fact, there will be continuum many ordinals with the same height – one for each strictly increasing function $f : \mathbb{N} \to \mathbb{N}$!) The order relations ought to classify ordinals by their stature.

We have to consider the question of what to make of ordinals like $\sqcup\{0, 0, 0, \ldots\}$. If we read ‘$\sqcup$’ as ‘supremum’, i.e., least (non-strict) upper bound, then that ordinal should have the same stature as 0. If we read ‘$\sqcup$’ as ‘least strict upper bound, then it should have the same stature as 1, i.e., 0^+. We will have to face up to the somewhat unpleasant fact that it will not be decidable whether an ordinal has (the same) stature (as) 0, or a successor ordinal.

There are many, many paths one can take starting at this point. One can for example contrive that $\sqcup$ ordinals always have limit height, and this can be done in various ways. One might regard the Brouwer ordinals, our basic datatype, as too broad: we could pick out from ‘raw’ ordinals those that are ‘good’ in the sense that the sequence $\alpha_0, \alpha_1, \alpha_2, \ldots$ to which $\sqcup$ is applied should be strictly increasing in the *structural* order, and that this ‘goodness’ should obtain ‘all the way down’. There are various other tricks. . . .

2.2 A type-theoretical approach

Here we’ll look at an approach to defining order relations on the ordinals in Martin-Löf type theory. It may not be, in the end, completely satisfactory: that needs to be investigated. It is at least likely that some of the techniques may be useful for a solution.

We begin with a useful distinction between two ways of representing the 'power' of a set/type in type theory. (In impredicative set theory, the power of a set is a set, so the operation is called 'powerset'.)

$$\begin{aligned}
&\mathit{Fam}, \mathit{Pow} : \mathsf{Type} \to \mathsf{Type} \\
&\mathit{Fam}\, A \triangleq (\Sigma\, I : \mathsf{Set})\, I \to A \\
&\mathit{Pow}\, A \triangleq A \to \mathsf{Set}
\end{aligned}$$

Consider *Fam* first. If A is a type (which may be large, such as for example the type Set of sets, i.e., small types), an element of $\mathit{Fam}\, A$ is a set $I : \mathsf{Set}$, together with an A-valued function $a : I \to A$ with domain I.

$$\langle I, a\rangle : \mathit{Fam}\, A$$

Think of this data as giving a 'subset' of A by exhaustive enumeration (possibly with repetitions), that one might write in the 'replacement-style' set-theoretic notation $\{a\, i \,|\, i : I\}$.

Now consider *Pow*. If A is a type (which may, again, be large), an element of $\mathit{Pow}\, A$ is simply a set-valued function defined on A. In line with the Curry-Howard correspondence, we may think of Set as a space of generalised truth values or propositions. Then an element of $\mathit{Pow}\, A$ is simply a truth-valued function, or characteristic function.

$$P : A \to \mathsf{Set}$$

Think of P as giving a 'subset' of A consisting of those $a : A$ such that $P\, a$ is inhabited. We might write this in the 'separation-style' set-theoretic notation $\{a : A \,|\, P\, a\}$.

Note that even if the argument type A is small, that value $\mathit{Fam}\, A$ or $\mathit{Pow}\, A$ is large. It is built on top of Set. Note moreover that $\mathit{Fam}\, A$ is covariant (positive) in A, whereas $\mathit{Pow}\, A$ is contravariant (negative, in that it contains A to the left of an arrow). Indeed, *Fam* is the basis of the covariant powerset functor (that on morphisms $f : A \to A'$ sends $\langle I, a\rangle : \mathit{Fam}\, A$ to $\langle I, f \cdot a\rangle : \mathit{Fam}\, A'$), whereas *Pow* is the basis of the contravariant one (that on morphisms $f : A \to A'$ sends $P : \mathit{Pow}\, A'$ to $P \cdot f : \mathit{Pow}\, A$.

When A is small, we can pass reasonably freely, by a kind of 'somersault', between these two notions of subset.

- Given $\langle I, f\rangle : \mathit{Fam}\, A$, its *Pow*-version is $\{a : A \,|\, (\Sigma\, i : I)\, f(i) =_A a\}$, where $(=_A)$ denotes 'the' equality relation on A. (Alternatively, if $\{a\} : \mathit{Pow}\, A$ denotes the singleton predicate true of just the element $a : A$, then we can represent this *Pow*-version as the union $\cup_{i:I}\{f\, i\}$.

- Given $P : \mathit{Pow}\, A$, its *Fam*-version is $\langle(\Sigma\, a : A)\, P\, a, (\lambda\, \langle a, \_\rangle)\, a\rangle$. That is to say, we take for the index set the Σ-type $(\Sigma\, a : A)\, P\, a$, and for the indexing function the projection of an ordered pair onto its first coordinate. (Note that the same $a : A$ gets 'repeated', once for each proof of $a : A$.

Note however that if A is not a set, the first half of this somersault is blocked, because only Sets come equipped with an equality relation. Moreover, the second half as well is blocked, because then the domain of the projection function is not a set.

After this detour, we return to the matter at hand, namely order relations between Brouwer ordinals. First we define a structural strict order relation, of type $\Omega \to \mathit{Fam}\,\Omega$.

We can define the transitive strict predecessor relation on Ω as follows. We define a set valued function $Pd^{+}(\alpha)$, and an Ω-valued function $\alpha[\_] : (\Pi\,\alpha : \Omega)\,Pd^{+}(\alpha) \to \Omega$ both by recursion on α.

$$\begin{array}{ll}
Pd^{+}(0) = N_0 & \\
Pd^{+}(\alpha^{+}) = 1 + Pd^{+}(\alpha) & (\alpha^{+})[o]^{+} = \alpha \\
 & (\alpha^{+})[s(t)]^{+} = \alpha[t]^{+} \\
Pd^{+}(\sqcup_i \alpha_i) = (\Sigma\, n \in \mathbb{N})\, Pd^{+}(\alpha_n) & (\sqcup_i \alpha_i)[\langle n, t\rangle]^{+} = \alpha_n[t]^{+}
\end{array}$$

(In the clauses for $\alpha^{+}[\_]$, I have used o and s as constructors for the respective arms of the disjoint sum $1 + Pd^{+}(\alpha)$. And I apologise for the clash of notation between successor α^{+} and transitive closure Pd^{+}, $\_[\_]^{+}$!)

Note: $pd^{+} \triangleq (\lambda\,\alpha)\,\langle Pd^{+}(\alpha), \alpha[\_]^{+}\rangle : \Omega \to \mathit{Fam}(\Omega)$. Now the universe of index sets contains $theemptysetN_0$, is closed under set-successor, and countable disjoint union.

Essentially, the set $Pd^{+}(\alpha)$ consists of the paths down from α that cross at least one successor ordinal.

Next we turn to a 'statural' non-strict order relation, of type $\Omega \to \mathit{Pow}\,\Omega$.

Here we are trying to capture the intuition that ever downward transition in α can be matched by a downward transition of β. We do not assume that a step from $\sqcup_i \alpha_i$ to α_n is necessarily a step down. (In other words, we interpret $\sqcup_i \alpha_i$ as a supremum, or least upper bound.) We write this

$$\alpha \preceq \beta$$

We define $\alpha \preceq \beta$ by recursion on α (into a universe containing a singleton set N_1, and closed under certain forms of Σ, and countable Π):

$$\begin{array}{ll}
0 \preceq \beta & \triangleq N_1 \\
\alpha^{+} \preceq \beta & \triangleq (\Sigma\, t \in Pd^{+}(\beta))\, \alpha \preceq \beta[t]^{+} \\
\sqcup_i \alpha_i \preceq \beta & \triangleq (\Pi\, n \in \mathbb{N})\, \alpha_n \preceq \beta
\end{array}$$

Now one can define 'equality of stature' (a very extensional relation) by:

$$\alpha \simeq \beta \qquad \triangleq \qquad (\alpha \preceq \beta) \times (\beta \preceq \alpha)$$

Another useful relation is the strict external relation.

$$\alpha \prec \beta \triangleq (\Sigma\, t : Pd^{+}(\beta))\, \alpha \preceq \beta[t]^{+}$$

What I hope you have gathered from the above is that a complete treatment of the order and equality relations between countable ordinals (considered as inductively defined data structures) is rather delicate. (This is often the case in the constructive treatment of any interesting mathematical structure.)

2.3 Order types and von Neumann ordinals

In view of the difficulties indicated in the last section of developing the theory of order relations on the Brouwer ordinals constructively, we now but to a more intuitive pictorially based and informal point of view that corresponds more closely to the set-theoretic notion.

Let us consider structures $\langle A, \leq \rangle$ where $\leq$ is a total order on A (a partial order such that all elements are comparable, in the sense $(\forall a, b : A) a < b \vee a = b \vee a > b$, where $<$ is the irreflexive part of $\leq$). These form a category, in which the morphisms from $(\langle A, \leq_A \rangle$ to $\langle B, \leq_B \rangle$ are functions on the underlying sets $f : A \rightarrow B$ that preserve the strict relations in the sense $\forall a, b : A) a < b \rightarrow f\,a < f\,b$. We shall be most interested in the full subcategory in which the structures are *well-ordered*, i.e., in which all descending chains are finite. (Warning: this is not a very satisfactory definition constructively.) An ordinal (in one sense) is an *isomorphism class* of such structures – a collection of ordered structures which are pairwise isomorphic. The isomorphism class of $\langle A, \leq \rangle$ is sometimes called the *order-type* of $\langle A, \leq \rangle$. It really is a *class* (i.e., belongs to the set-theoretic counterpart of `Type` – something *large*).

How should we quantify over all ordinals? That would seem to require some kind of set-theory with class variables, or even (as soon as we start to deal with classes of ordinals and such things) variables of higher order. There are indeed such systems (Gödel-Bernays, Morse-Kelly,. . .) though they are not particularly well-known, or for that matter attractive. The situation is saved by choosing a particular, canonical representation of each isomorphism class. These representations are usually called *von Neumann* ordinals, who introduced them (in 1929).

An ordinal (in another sense) is a transitive set well-ordered by the set-theoretic membership relation $\in$. (A set x is transitive if it satisfies $(\forall y, z) z \in y \rightarrow y \in x \rightarrow z \in x$. Actually, in the presence of the foundation/regularity axiom of set-theory – which says that the relation $\in$ is wellfounded on the class of sets – it is equivalent to define a set to be an ordinal if it is a transitive set all of whose elements are transitive. See for example `http://en.wikipedia.org/wiki/Ordinal_number`.)

Fact: Every well-ordering $\langle A, < \rangle$ is order isomorphic to an ordinal $\langle \alpha, \in \rangle$.

As you will probably know, the finite ordinals (the natural numbers) in the von Neumann representation look as follows:

$$\begin{aligned}
&0 = \emptyset = \{\},\\
&1 = \{0\} = \{\{\}\},\\
&2 = \{0, 1\} = \{\{\}, \{\{\}\}\},\\
&3 = \{0, 1, 2\} = \{\{\}, \{\{\}\}, \{\{\}, \{\{\}\}\}\}\\
&\ldots
\end{aligned}$$

If you are a programmer, you'll notice that these seem to be something like particularly well behaved Rose trees. (Of course, Rose trees are made of of lists and not sets.)

In general each von Neumann ordinal is nothing but the set of its predecessors. It is not difficult to see that if α is a ordinal, so is $\alpha \cup \{\alpha\}$, which we can write α^+, and this is the least ordinal greater than α. Moreover α is the greatest ordinal smaller that α^+, or its immediate predecessor. Not every ordinal, however, has an immediate predecessor: for example 0 and ω. The ordinals that have immediate predecessors are called *successors*, and the non-zero ordinals that do not are called *limits*.

Purely for amusement,

EXERCISE 3 *Define (in Haskell) the infinite stream of 'von Neumann Rose trees'.*

Forgetting von Neumann ordinals, and remembering the definitions of $+$, $\times$ and $\uparrow$ given above, let us give explicit constructions on well-ordered sets that correspond to each of those operations. (There will be a wrinkle in the case of exponentiation.)

Addition Suppose $\langle A, \leq_A \rangle$ and $\langle B, \leq_B \rangle$ are totally ordered collections. (We actually don't need them to be well-ordered, or even sets.) We define the ordered sum $\langle A, \leq_A \rangle + \langle B, \leq_B \rangle$ to have underlying set $A + B$ (the disjoint union of A and B, with constructors i and j respectively. As for the order

$$\begin{array}{ll} i(a) \leq i(a') & = a \leq_A a' \\ i(a) \leq j(b) & = \text{true} \\ j(b) \leq i(a) & = \text{false} \\ j(b) \leq j(b') & = b \leq_B b' \end{array}$$

The definition amounts to putting a copy of the second *after* a copy of the first. It is easy to see that the ordered sum is well-ordered just in case both summands are well-ordered. It is not difficult to see that

$$\begin{array}{l} \alpha + 0 = \alpha \\ \alpha + \beta^+ = (\alpha + \beta)^+ \\ \alpha + \lambda = \sqcup_{\beta < \lambda}(\alpha + \beta) \end{array}$$

where (as is traditional) in the last clause λ stands for a limit (and the equality sign means order isomorphism).

EXERCISE 4 *Prove the following:*

$$\begin{array}{l} \alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma \\ \alpha + 0 = \alpha = 0 + \alpha \end{array}$$

So $0, +$ forms a monoid. It is not though commutative. For example $1 + \omega = \omega$, while $\omega < \omega + 1$. (At this point, one should mention the so-called 'natural' or Hessenberg sum and product. These are commutative operations, with better distributivity properties than Cantor's operations, with which they sometimes – but only sometimes – coincide. The Hessenberg natural sum plays a role here and there in ordinal theoretic proof theory. There are also connections with Conway's 'surreal arithmetic'. I encourage you to look up the definitions in Wikipedia, where there is a good explanation.)

Other noteworthy facts and non-facts are the following:

$$\begin{aligned}
&\alpha < \beta \Longrightarrow \gamma + \alpha < \gamma + \beta\\
&\alpha < \beta \Longrightarrow \alpha + \gamma \leq \beta + \gamma\\
&\alpha + \beta = \alpha + \gamma \Longrightarrow \beta = \gamma
\end{aligned}$$

whereas, we do *not* have $\alpha + \beta = \gamma + \beta \Longrightarrow \alpha = \gamma$ Moreover, although $\alpha + \beta$ is continuous in its second argument, we do *not* have $\lambda + \alpha = \sqcup_{\beta<\lambda}(\beta + \alpha)$ – addition is not continuous in its first argument.

Multiplication Suppose $\langle A, \leq_A\ \rangle$ and $\langle B, \leq_B\ \rangle$ are totally ordered collections. (We actually don't need them to be well-ordered, or even sets.) We define the ordered product $\langle A, \leq_A\ \rangle \times \langle B, \leq_B\ \rangle$ to have underlying set $A \times B$ (the (binary) cartesian product of A and B, with constructor $\langle \_, \_ \rangle$ respectively. As for the order

$$\langle a_1, b_1\rangle \leq \langle a_2, b_2\rangle \quad = b_1 <_B b_2 \vee (b_1 =_B b_2 \wedge a_1 \leq_A a_2)$$

This order is sometimes called the *reverse* lexicographic ordering, since it corresponds to the order in which words appear in a Persian dictionary. It amounts to putting a copy of A in place of each element of a copy of B, with the obvious tagging. (The reason for the reverse lexicography is to make the arithmetic neater. One wants to have properties like $\alpha^{\beta+\gamma} = \alpha^\beta \times \alpha^\gamma$, rather than repulsive isotopes like $\alpha^{\beta+\gamma} = \alpha^\gamma \times \alpha^\beta$. In fact in Cantor's original paper in 1983, he used the superficially more natural forward lexicographic ordering, and switched after 4 years when he had figured out what was involved.

It is easy to see that the ordered product is well-ordered just in case both factors are well-ordered. It is not difficult to see that

$$\begin{aligned}
&\alpha \times 1 = \alpha\\
&\alpha \times \beta^+ = (\alpha \times \beta) + \alpha\\
&\alpha \times \lambda = \sqcup_{\beta<\lambda}(\alpha \times \beta)
\end{aligned}$$

EXERCISE 5 *Prove the following:*

$$\begin{aligned}
&\alpha \times (\beta \times \gamma) = (\alpha \times \beta) \times \gamma\\
&\alpha \times 1 = \alpha = 1 \times \alpha
\end{aligned}$$

So $1, \times$ forms a monoid. It is not though commutative. For example $2 \times \omega = \omega$, while $\omega < \omega \times 2$.

Fact:

$$\begin{aligned}
&\alpha \times (\beta + \gamma) = (\alpha \times \beta) + (\alpha \times \gamma)\\
&\alpha \times 0 = 0
\end{aligned}$$

So $(\alpha\times)$ distributes over the monoid $0, +$.

EXERCISE 6 *Find a counterexample to* $(\beta + \gamma) \times \alpha = (\beta \times \alpha) + (\gamma \times \alpha)$.

It *happens* to be the case that $0 \times \alpha = 0$. But as we shall see later, when we come to define ordinals in Gödel's system T, this is in some sense 'a mere accident'.

Other noteworthy facts and non-facts are the following:

$$\begin{array}{ll} \alpha < \beta \Longrightarrow \gamma \times \alpha < \gamma \times \beta & \text{provided } \alpha > 0 \\ \alpha \leq \beta \Longrightarrow \alpha \times \gamma \leq \beta \times \gamma & \\ \alpha \times \beta = \alpha \times \gamma \Longrightarrow \beta = \gamma & \text{provided } \alpha > 0 \end{array}$$

Just like addition, although multiplication is continuous in its second argument, it is not continuous in the first.

Exponentiation . (This is considerably more tricky than the case of addition and multiplication.) Suppose $\langle A, \leq_A \rangle$ and $\langle B, \leq_B \rangle$ are totally ordered collections and A has least element 0. We define the ordered exponential $\langle A, \leq_A \rangle \uparrow \langle B, \leq_B \rangle$ to have as its underlying set $A \times B$ the subset of A^B (i.e., $B \rightarrow A$) consisting of functions that are non-zero for all but finitely many arguments (Sometimes called functions with finite support). As for the order

$$f \leq g \quad = f = g \vee (f \neq g \wedge f\, b_m <_A g\, b_m \text{ where } b_m = \max\{b \in B \mid f\, b \neq g\, b\})$$

This definition is (at least, on the face of it) rather non-constructive, and admittedly rather hard to motivate. In practice, there are only two bases α for which ordinal exponentiation $\alpha \uparrow \beta$ 'really matters', namely 2 and ω, and these are fairly easy to understand.

Take the case 2^β (where β is the order type of $\langle B, \leq_B \rangle$). Then a function with finite support is essentially a finite subset of B, and we can identify such a finite subset with the list of its elements taken in strictly decreasing order. Now order such sequences in the (forward) lexicographic order. That's 2^β!

Now take the case ω^β. Then a function with finite support is essentially a (finite) multiset or bag of elements of B. We can think of such a bag as a finite set of ordered pairs $\langle b, n \rangle$ where $b \in B$, and $n > 0$ is a natural number, the multiplicity of b. We can identify such a set of ordered pairs with the sequence in which such pairs are listed in descending order of their first component (the 'b'). Now order such descending sequences lexicographically, where the pairs $\langle b, n \rangle$ are in the *reverse* lexicographic order of pairs. That's ω^β!

That may still seem quite complicated. I believe though that you will understand the matter if (in the case ω^β you think of ordinary polynomials like $x^5.3 + x^2.19 + 23$, with exponents and coefficients which are natural numbers. Think of such a polynomial p as arranged in descending order of exponents, and as representing a one-place function on the natural numbers $(x \mapsto p(x)) : \mathbb{N} \rightarrow \mathbb{N}$. There is a fairly natural ordering of such polynomials, in which polynomials are ordered by the coefficient of the greatest exponent at which they differ. Why is that natural? Because it reflects the order of *eventual dominance* between the functions that the polynomials represent.

$$f <_{ed} g \quad \triangleq \quad (\exists n)(\forall m) f(n + m) < g(n + m)$$

The ordering of polynomials is, if you think it through, ω^ω: the base ω because the coefficients are natural numbers, and the exponent ω because the exponents are natural numbers as well. (The case 2^β is quite similar, except that the coefficients are less than 2.)

What is really going on with the functions of finite support is this: such a function $B \to A$ which is zero for all but a finite number of arguments is a formal polynomial, with finitely many identically zero summands. Where the function has a non-zero value (in A), that value is the coefficient for the exponent (from B) which is its argument. We order such formal polynomials by the natural order, corresponding to eventual dominance.

In the cases of addition and multiplication, our constructions made perfect sense even when the ordered sets involved were not well-orderings, or even total. In the case of exponentiation, I am not sure. If the exponent structure is not well founded, it may be that the functions with finite support should be replaced by functions which are zero except on elements of some (totally ordered, descending) *chain* in B.

We had better finish up by listing some of the 'algebraic' properties of exponentiation.

First are 'laws of exponents'

$$\begin{array}{ll}
\alpha \uparrow (\beta \times \gamma) & = (\alpha \uparrow \beta) \uparrow \gamma \\
\alpha \uparrow 1 & = \alpha \\
\alpha \uparrow (\beta + \gamma) & = (\alpha \uparrow \beta) \times (\alpha \uparrow \gamma) \\
\alpha \uparrow 0 & = 1
\end{array}$$

It *happens* to be the case that $1^\alpha = 1$, and that (for $\alpha > 0$) $0^\alpha = 0$. (This is perhaps a strange thing to say, but my reasons will emerge later on.)

It is *not* the case that $(\alpha \times \beta) \uparrow \gamma = (\alpha \uparrow \gamma) \times (\beta \uparrow \gamma)$.

Some properties of $\uparrow$ that involve the order relations are these.

$$\begin{array}{ll}
\alpha < \beta \Longrightarrow \gamma \uparrow \alpha < \gamma \uparrow \beta & \text{provided } \alpha > 1 \\
\alpha \uparrow \beta = \alpha \uparrow \gamma \Longrightarrow \beta = \gamma & \text{provided } \alpha > 1 \\
\alpha \leq \beta \Longrightarrow \alpha \uparrow \gamma \leq \beta \uparrow \gamma &
\end{array}$$

EXERCISE 7 *Verify the following.*

$$\begin{array}{ll}
2^\omega & = \omega \\
2^{\omega+1} & = \omega \times 2 \\
2^{\omega \times 2} & = \omega^2 \\
2^{\omega^2} & = \omega^\omega \\
2^{\omega^\omega} & = \omega^{\omega^\omega}
\end{array}$$

2.4 Cantor normal form

The ordinal 0 is the empty sum. It so happens that when β is at least 2, that every ordinal α can be uniquely decomposed into a finite sum.

$$\begin{array}{l} \alpha = \beta^{\alpha_1} \times \delta_1 + \cdots + \beta^{\alpha_k} \times \delta_k \\ \alpha \geq \alpha_0 > \ldots > \alpha_k \geq 0 \\ 0 < \delta_0, \ldots \delta_k < \beta \end{array}$$

The proof is quite tricky in our second number class. The idea is to exploit the following operations which are right adjoint to exponentiation, multiplication and addition.

$$\begin{array}{ll} \beta^\gamma \leq \alpha & \equiv \gamma \leq \log_\beta \alpha \\ \beta \times \gamma \leq \alpha & \equiv \gamma \leq \alpha/\beta \\ \beta + \gamma \leq \alpha & \equiv \gamma \leq \alpha - \beta \end{array}$$

The first ($\log_\beta$) is defined only for β at least 2, α at least 1. The second $((/\beta))$ is defined only for β at least 1. The third $((-\beta)\alpha)$ is defined only for $\alpha \geq \beta$.

To define these constructively is extremely subtle, but we can pretend that they exist. The difficult principle that for any normal function, there is a greatest ordinal such that $f\alpha \leq \beta$. No doubt, but we may not be able to compute it.

One defines

$$\begin{array}{llll} \alpha_0 = \log_\beta \alpha & \alpha_1 = \log_\beta \gamma_0 & & \alpha_k = \log_\beta \gamma_k \\ \delta_0 = \alpha/(\beta^{\alpha_0}) & \delta_1 = \alpha/(\beta^{\alpha_1}) & \cdots & \delta_k = \alpha/(\beta^{\alpha_k}) \\ \gamma_0 = \alpha - (\beta^{\alpha_0} \times \delta_0) & \gamma_1 = \alpha - (\beta^{\alpha_1} \times \delta_1) & & \gamma_k = \alpha - (\beta^{\alpha_k} \times \delta_k) = 0 \end{array}$$

We usually consider Cantor normal form only to the base ω.

The manipulation of expressions in Cantor normal form is quite fun. It pivots on the fact that $\omega^\alpha + \omega^\beta = \omega^\beta$ when $\alpha < \beta$.

So if $\alpha = \omega^{\alpha_1} \times n_1 + \cdots + \omega^{\alpha_k} \times n_k$ $\beta = \omega^{\beta_1} \times n_1 + \cdots + \omega^{\beta_l} \times n_l$ then in $\alpha + \beta$, β will 'eat' some suffix of the representation of α.

EXERCISE 8 *If you have lots of time, consider how to rewrite arbitrary expressions built up from* 0, 1 *and* ω *by* $+$, $\times$ *and* $\uparrow$ *to expressions in Cantor normal form (to the base* ω*).*

2.5 Normal functions

A function $f : \Omega \to \Omega$ is said to be normal if it is strictly increasing and continuous. For example $1 + \alpha$ is normal, but $\alpha + 1$ is not.

All functions $\alpha \mapsto f^\alpha \ldots$ are continuous. If f is continuous, then f^ω is a closure operator. If f is a closure operator, then $(f \cdot (+1))^\alpha 0$ is a normal function of α. If f is normal, the function

$$f' : \alpha \mapsto (f^\omega \cdot (+1))^\alpha (f^\omega 0)$$

is also normal, and enumerates the fixed points of f.

This step from a normal function to the function which enumerates its fixed points is called Veblen's derivative operation. We write it ∇. Given a normal function $f : \Omega \to \Omega$, one can define a two place function

$$\begin{array}{l}
\phi : \Omega \to \Omega \to \Omega \\
\phi_0 = f \\
\phi_{\alpha^+} = \nabla(\phi_\alpha) \\
\phi_{\sqcup_i \alpha_i}) = \text{enumerates } \ \{\beta : \Omega \,|\, (\Pi\, n : \omega)\, \phi_{\alpha_n} \beta = \beta\}
\end{array}$$

One can write out the definition of ϕ by recursion on its two arguments.

$$\begin{array}{llllll}
\phi_0 & \triangleq f & & & & \\
\phi_{\alpha^+} 0 & \triangleq (\phi_\alpha)^\omega 0 & \phi_{\alpha^+}(\beta^+) & \triangleq (\phi_\alpha)^\omega(\phi_{\alpha^+}\beta + 1) & \phi_{\alpha^+} \sqcup_i \beta_i & \triangleq \sqcup_i \phi_{\alpha^+}(\beta_i) \\
\phi_{\sqcup_i \alpha_i} 0 & \triangleq \sqcup_i \phi_{\alpha_i} 0 & \phi_{\sqcup_i \alpha_i} \beta^+ & \triangleq \sqcup_i \phi_{\alpha_i}(\phi_{\sqcup_i \alpha_i} \beta + 1) & \phi_{\sqcup_i \alpha_i} \sqcup_i \beta_i & \triangleq \sqcup_i \phi_{\sqcup_j \alpha_j}(\beta_i)
\end{array}$$

The Veblen hierarchy over $1 + \alpha$ is as follows:

$$\begin{array}{l}
1 + \alpha \\
\omega + \alpha \\
\omega^2 + \alpha
\end{array}$$

At the βth level we have $\phi_\beta \alpha = \omega^\alpha + \beta$.

The function ω^α is normal in the argument α. We usually erect the Veblen hierarchy over the function ω^α.

$\phi_0(\alpha) = \omega^\alpha$	ordinals closed under addition
$\phi_1(\alpha) = \varepsilon_\alpha$	ordinals closed under $\alpha \mapsto \omega^\alpha$
$\phi_2(\alpha)$	critical ε numbers: $\alpha = \varepsilon_\alpha$

The function $\phi\, \alpha\, 0$ is normal in α, and is usually written Γ_α. One can take this as the basis of a new Veblen hierarchy, and iterate the process – this leads to the so-called "3-place" Veblen function. One can iterate the idea of new argument places. This leads to the idea of transfinitely indexed argument places, and this to the idea of Schütte's 'Klammer-symbolen'. In fact these were anticipated by Veblen himself, in a publication from 1908. (One hundred years ago!)

The least ordinal which closed under the finite-place Veblen functions is sometimes called the 'little' Veblen number, and you may find it written $\phi_{\Omega^\omega}(0)$.

The least ordinal which closed under the Veblen functions of transfinite arity (using arities obtainable 'from below', or autonomously) is sometimes called the 'big' Veblen number, and you may find it written $\phi_{\Omega^\Omega}(0)$.

Both Veblen numbers (the little one and the big one) correspond to certain versions of Martin-Löf type theory, without W-types, but extended with rather strong universe axioms. The investigation of such systems sometimes goes under the name 'metapredicativity'.

The 'next' celebrated ordinal is the famous 'Bachmann-Howard' ordinal, that you will often find written $\phi_{\epsilon_{\Omega+1}}(0)$, which measures the strength of a host of interesting systems. Bachmann was the first person to devise a system of ordinal notations to describe the ordinal, and Howard was the first proof-theorist to investigate a system with that proof-theoretic ordinal: essentially (I think) Gödel's T, but extended by the countable Brouwer ordinals. It is an interesting challenge to devise universe principles that when added to type-theory without W-types attains the strength of the Bachmann-Howard ordinal.

2.6 TODO: Ordinal representation systems

3 Lower Bounds

We work in a type theory closed under →, and that satisfies the η-rule. Moreover it should contain (a symbol for) the type of natural numbers $\mathbb{N}$, with 0, (+1), and an iteration functional

$$\begin{array}{ll} \mathcal{I} : (X \to X) \to N \to X \to X & \\ \mathcal{I}\, f\, 0 & = id \\ \mathcal{I}\, f\, (n+1) & = f \cdot \mathcal{I}\, f\, n \end{array}$$

3.1 Church ordinals

We begin with a notion of 'Church ordinal'. Such a thing is a term —t— with 4 free variables:

$$X : \mathsf{Set}, z : X, s : X \to X, l : (N \to X) \to X \vdash t[X, z, s, l] : X$$

For example:

$$\begin{array}{ll} t_0[X, z, s, l] & = z \\ t_1[X, z, s, l] & = s\, z \\ t_2[X, z, s, l] & = s\, (s\, z) \\ t_\omega[X, z, s, l] & = l\, (n \mapsto \mathcal{I}\, s\, n\, z) \end{array}$$

Suppose a and b are Church ordinals, then

$$\begin{array}{ll} t_{a+b}[X, z, s, l] & = t_b[X, t_a[X, z, s, l], s, l] \\ t_{a \times b}[X, z, s, l] & = t_b[X, z, x \mapsto t_a[X, x, s, l], l] \\ t_{a \uparrow b}[X, z, s, l] & = t_b[X \to X, \\ & \quad s, \\ & \quad f\, x \mapsto t_a[X, x, f, l], \\ & \quad g\, x \mapsto l(n \mapsto g\, n\, x) \\ & \quad]\, z \end{array}$$

But what justifies this annotation? The lemmas below state that certain algebraic equations hold definitionally; these equations include the defining equations of $+, \times, \uparrow$.

For abbreviation below, we define $\mathcal{L}\, l = g \mapsto l \cdot \mathit{flipg}$.

Some of the expected laws hold:

$$\begin{aligned}
\llbracket a+0\rrbracket[X,z,s,l] &= \llbracket 0\rrbracket[X,\llbracket a\rrbracket[X,z,s,l],s,l]\\
&= \llbracket a\rrbracket[X,z,s,l]\\
\llbracket 0+a\rrbracket[X,z,s,l] &= \llbracket a\rrbracket[X,\llbracket 0\rrbracket[X,z,s,l],s,l]\\
&= \llbracket a\rrbracket[X,z,s,l]\\
\llbracket a+(b+c)\rrbracket[X,z,s,l] &= \llbracket b+c\rrbracket[X,\llbracket a\rrbracket[X,z,s,l],s,l]\\
&= \llbracket c\rrbracket[X,\llbracket b\rrbracket[X,\llbracket a\rrbracket[X,z,s,l],s,l],s,l]\\
\llbracket (a+b)+c\rrbracket[X,z,s,l] &= \llbracket c\rrbracket[X,\llbracket a+b\rrbracket[X,z,s,l],s,l]\\
&= \llbracket c\rrbracket[X,\llbracket b\rrbracket[X,\llbracket a\rrbracket[X,z,s,l],s,l],s,l]
\end{aligned}$$

Figure 1: $0,+$ forms a monoid.

$$\begin{aligned}
\llbracket a\times 1\rrbracket[X,z,s,l] &= \llbracket 1\rrbracket[X,z,x\mapsto\llbracket a\rrbracket[X,x,s,l],l]\\
&= \llbracket a\rrbracket[X,z,s,l]\\
\llbracket 1\times a\rrbracket[X,z,s,l] &= \llbracket a\rrbracket[X,z,x\mapsto\llbracket 1\rrbracket[X,x,s,l],l]\\
&= \llbracket a\rrbracket[X,z,x\mapsto sx,l]\\
&= \llbracket a\rrbracket[X,z,s,l]\\
\llbracket a\times(b\times c)\rrbracket[X,z,s,l] &= \llbracket b\times c\rrbracket[X,z,s',l]\\
&\quad\text{where } s' = x\mapsto\llbracket a\rrbracket[X,x,s,l]\\
&= \llbracket c\rrbracket[X,z,x\mapsto\llbracket b\rrbracket[X,x,s',l],l]\\
&= \llbracket c\rrbracket[X,z,x\mapsto\llbracket b\rrbracket[X,x,y\mapsto\llbracket a\rrbracket[X,y,s,l],l],l]\\
\llbracket (a\times b)\times c\rrbracket[X,z,s,l] &= \llbracket c\rrbracket[X,z,s',l]\\
&\quad\text{where } s' = x\mapsto\llbracket a\times b\rrbracket[X,x,s,l]\\
&\quad\phantom{\text{where } s'} = x\mapsto\llbracket b\rrbracket[X,x,y\mapsto\llbracket a\rrbracket[X,y,s,l],l]\\
&= \llbracket c\rrbracket[X,z,x\mapsto\llbracket b\rrbracket[X,x,y\mapsto\llbracket a\rrbracket[X,y,s,l],l],l]
\end{aligned}$$

Figure 2: $1,\times$ forms a monoid

LEMMA 1 $0,+$ *forms a monoid.*

$$\begin{aligned}
&\llbracket 0+a\rrbracket = \llbracket a\rrbracket = \llbracket a+0\rrbracket\\
&\llbracket a+(b+c)\rrbracket = \llbracket (a+b)+c\rrbracket
\end{aligned}$$

Proof by the calculations in figure 1

LEMMA 2 $1,\times$ *forms a monoid.*

$$\begin{aligned}
&\llbracket 1\times a\rrbracket = \llbracket a\rrbracket = \llbracket a\times 1\rrbracket\\
&\llbracket a\times(b\times c)\rrbracket = \llbracket (a\times b)\times c\rrbracket
\end{aligned}$$

Proof by calculations in figure 2

LEMMA 3 $(a\times)$ *commutes with the* $0,+$ *monoid.*

$$\begin{aligned}
\llbracket a\times 0\rrbracket &= \llbracket 0\rrbracket\\
\llbracket a\times(b+c)\rrbracket &= \llbracket a\times b+a\times c\rrbracket
\end{aligned}$$

$$
\begin{aligned}
\llbracket a \times 0 \rrbracket[X,z,s,l] &= \llbracket 0 \rrbracket[X,z,x \mapsto a[X,x,s,l],l] \\
&= \llbracket 0 \rrbracket[X,z,s,l] \\
\llbracket a \times (b+c) \rrbracket[X,z,s,l] &= \llbracket b+c \rrbracket[X,z,x \mapsto \llbracket a \rrbracket[X,x,s,l],l] \\
&= \llbracket c \rrbracket[X,\llbracket b \rrbracket[X,z,x \mapsto \llbracket a \rrbracket[X,x,s,l],l],x \mapsto \llbracket a \rrbracket[X,x,s,l],l] \\
\llbracket a \times b + a \times c \rrbracket[X,z,s,l] &= \llbracket a \times c \rrbracket[X,\llbracket a \times b \rrbracket[X,z,s,l],s,l] \\
&= \llbracket c \rrbracket[X,\llbracket a \times b \rrbracket[X,z,s,l],x \mapsto \llbracket a \rrbracket[X,x,s,l],l] \\
&= \llbracket c \rrbracket[X,\llbracket b \rrbracket[X,z,x \mapsto \llbracket a \rrbracket[X,x,s,l],l],x \mapsto \llbracket a \rrbracket[X,x,s,l],l]
\end{aligned}
$$

Figure 3: $(a\times)$ commutes with the $0,+$ monoid

$$
\begin{aligned}
\llbracket a \uparrow 0 \rrbracket[X,z,s,l] &= \llbracket 0 \rrbracket \; \begin{array}[t]{ll} [& X \to X, \\ & s, \\ & f\,x \mapsto \llbracket a \rrbracket[X,x,f,l], \\ & \mathcal{L}\,l \\ ] & z \end{array} \\
&= s\,z \\
&= \llbracket 1 \rrbracket[X,z,s,l] \\
\llbracket a \uparrow 1 \rrbracket[X,z,s,l] &= \llbracket 1 \rrbracket \; \begin{array}[t]{ll} [& X \to X, \\ & s, \\ & f\,x \mapsto \llbracket a \rrbracket[X,x,f,l] \\ & \mathcal{L}\,l \\ ] & z \end{array} \\
&= (f\,x \mapsto \llbracket a \rrbracket[X,x,f,l])\,s\,z \\
&= \llbracket a \rrbracket[X,z,s,l]
\end{aligned}
$$

Figure 4: Exponentiation by 0 and 1

Proof by the calculations in figure 3

REMARK 1 *We don't get, modulo definitional equality, all the laws we might expect. For example:*

$$
\begin{aligned}
\llbracket 0 \times a \rrbracket &= \llbracket a \rrbracket[X,z,x \mapsto \llbracket 0 \rrbracket[X,x,s,l],l] \\
&= \llbracket a \rrbracket[X,z,id,l] \\
&\neq \llbracket 0 \rrbracket[X,z,s,l]
\end{aligned}
$$

LEMMA 4 *(Cantor's exponential calculus)*

$$
\begin{aligned}
\llbracket a \uparrow 0 \rrbracket &= \llbracket 1 \rrbracket \\
\llbracket a \uparrow (b+c) \rrbracket &= \llbracket (a \uparrow b) \times (a \uparrow c) \rrbracket \\
\llbracket a \uparrow 1 \rrbracket &= \llbracket a \rrbracket \\
\llbracket a \uparrow (b \times c) \rrbracket &= \llbracket (a \uparrow b) \uparrow c \rrbracket
\end{aligned}
$$

Proof by the calculations in figures 4, 5 and 6

PROP 1 *The definitions of $+, \times$ and $\uparrow$ are correct.*

$$
\begin{aligned}
\llbracket a \uparrow (b \times c) \rrbracket [X, z, s, l] &= \llbracket b \times c \rrbracket [X \to X, s, f\,x \mapsto \llbracket a \rrbracket [X, x, f, l] \mathcal{L}\,l]\, z \\
&= \llbracket c \rrbracket [X \to X, s, f \mapsto \llbracket b \rrbracket [X \to X, f, g\,y \mapsto \llbracket a \rrbracket [X, y, g, l], \mathcal{L}\,l], \mathcal{L}\,l]\, z \\
\llbracket (a \uparrow b) \uparrow c \rrbracket [X, z, s, l] &= \llbracket c \rrbracket [X \to X, s, f\,x \mapsto \llbracket a \uparrow b \rrbracket [X, x, f, l] \mathcal{L}\,l]\, z \\
&= \llbracket c \rrbracket [X \to X, s, f\,x \mapsto \llbracket b \rrbracket [X \to X, f, g\,y \mapsto \llbracket a \rrbracket [X, y, g, l], \mathcal{L}\,l] x \mathcal{L}\,l]\, z
\end{aligned}
$$

Figure 5: Exponentiation by ×

$$
\begin{aligned}
\llbracket a \uparrow (b + c) \rrbracket [X, z, s, l] &= \llbracket b + c \rrbracket [X \to X, s, \\
&\qquad\qquad f\,x \mapsto \llbracket a \rrbracket [X, x, f, l], \\
&\qquad\qquad \mathcal{L}\,l \\
&\quad]\, z \\
&= \llbracket c \rrbracket [X \to X, \llbracket b \rrbracket [X \to X, s, f\,x \mapsto \llbracket a \rrbracket [X, x, f, l], \mathcal{L}\,l], \\
&\qquad\quad f\,x \mapsto \llbracket a \rrbracket [X, x, f, l], \\
&\qquad\quad \mathcal{L}\,l \\
&\quad]\, z \\
\llbracket (a \uparrow b) \times (a \uparrow c) \rrbracket [X, z, s, l] &= \llbracket a \uparrow c \rrbracket [X, z, x \mapsto \llbracket a \uparrow b \rrbracket [X, x, s, l], l] \\
&= \llbracket c \rrbracket [X \to X, x \mapsto \llbracket a \uparrow b \rrbracket [X, x, s, l],\ ]\, z \\
&\qquad\quad f\,x \mapsto \llbracket a \rrbracket [X, x, f, l] \\
&\qquad\quad \mathcal{L}\,l \\
&= \llbracket c \rrbracket [X \to X, x \mapsto \llbracket b \rrbracket [X \to X, s, f\,x \mapsto \llbracket a \rrbracket [X, x, f, l], \mathcal{L}\,l]\, x, \\
&\qquad\quad f\,x \mapsto \llbracket a \rrbracket [X, x, f, l] \\
&\qquad\quad \mathcal{L}\,l \\
&\quad]\, z
\end{aligned}
$$

Figure 6: Exponentiation by +

Proof. By using some of the lemmas above we have

$$\begin{array}{lll}
a + 0 & = a, & \\
a + (b + 1) & = (a + b) + 1 & \\
& & \\
a \times 0 & = 0, & \\
a \times (b + 1) & = a \times b + a \times 1 & = a \times b + a \\
& & \\
a \uparrow 0 & = 1, & \\
a \uparrow (b + 1) & = a \uparrow b \times a \uparrow 1 & = a \uparrow b \times a
\end{array}$$

Moreover $+, \times, \uparrow$ commute with limits in their right arguments, so their definitions agree with the usual definitions of $+, \times, \uparrow$ by (transfinite) primitive recursion.

EXAMPLE 2 *We can put together the definitions of ω, $\uparrow$ and $+$, and simplify to get:*

$$[\![a + w \uparrow b]\!][X, z, s, l] = [\![b]\!][X \to X, s, \mathcal{L}\, l \cdot \mathcal{I}, \mathcal{L}\, l]([\![a]\!][X, z, s, l])$$

4 TODO: Lenses

4.1 The basic idea

A lens is a 5-tuple:

$$\begin{array}{l}
F : \mathit{Set} \to \mathit{Set} \\
Z : X \to (X \to X) \to ((N \to X) \to X) \to FX \\
S : X \to (X \to X) \to ((N \to X) \to X) \to FX \to FX \\
L : X \to (X \to X) \to ((N \to X) \to X) \to (N \to FX) \to FX \\
D : X \to (X \to X) \to ((N \to X) \to X) \to FX \to X
\end{array}$$

We can wrap this up more compactly. For example, let $BX = 1 + X + (N \to X)$. Then we can put Z, S and L together into a single “up” function $U : (BX \to X) \to B(FX) \to FX$, and rewrite the “down” function $D : (BX \to X) \to FX \to X$. Note that U is something quite different from a B-algebra morphism. That is the point! As for F, that need not even be a functor.

EXAMPLE 3 *Identity lens (obvious), composition of lenses:*

$$\begin{array}{ll}
FX & = F_1(F_2X) \\
Zzsl & = Z_1(Z_2zsl)(S_2zsl)(L_2zsl) \\
Szsl & = S_1(Z_2zsl)(S_2zsl)(L_2zsl) \\
Lzsl & = L_1(Z_2zsl)(S_2zsl)(L_2zsl) \\
Dzsl & = D_2zsl(D_1(Z_2zsl)(S_2zsl)(L_2zsl))
\end{array}$$

EXAMPLE 4 *Gentzen/Archimedes lens:*

$$\begin{aligned}
FX &= X \to X \\
Zzsl &= s \\
Szsl &= fx \mapsto l(n \mapsto \mathcal{I}fnx) \\
Lzsl &= \mathcal{L}l \\
Dzsl &= f \mapsto fz
\end{aligned}$$

DEF 1 *A lens* implements *a function —phi— on ordinals if*

$$(\phi alpha)Xzsl = Dzsl(\alpha(FX)(Zzsl)(Szsl)(Lzsl))$$

[We have to analyse what equality means here.]

Note: the Gentzen/Archimedes lens implements the function ω^{α}.

PROP 2 *Composition of lenses implements composition of the functions they implement. That is, if* $F_\phi, Z_\phi, S_\phi, L_\phi, D_\phi$ *implements* ϕ *and* $F_\psi, Z_\psi, S_\psi, L_\psi, D_\psi$ *implements* ψ *then* $F_\psi \cdot F_\phi, \ldots$ *implements* $\phi \cdot \psi$.

Given a sequence/stream of lenses t_n, we want to define its limit *limt*. However, that is beyond the scope of these notes.

4.2 TODO: Dependent lenses

coalgebra

5 Upper Bounds

In the previous section, I have tried to give an impression of some techniques one can use to set a lower bound on the provable ordinals of a type theory. Essentially, you write programs in the type theory that denote those ordinals. The core of it is to figure out what it is about the type-structure available to you that gives 'headroom' sufficient to construct the ordinals. This work sometimes leaves you with a strong impression that these programs *exhaust* the strength of the type theory. For a long time, I have tried to pin down that impression, and turn it into a theorem – and not just a theorem, but one with a simple, elegant and 'algebraic' proof. I have pursued some ideas (that have the flavour of logical relations), but so far without much success.

However nowadays it is almost routine to obtain upper bounds for the strength of type theories. How is it done?

Roughly speaking one carries out a realisability interpretation of the type theory in a classical, infinitary sequent calculus. By an *infinitary* system, I mean a system that contains rules that have infinitely many premises. Such a system is not a formal system, in the ordinary sense of the term, as it is no longer decidable whether a particular step in a proof has the form of a rule. A proof in such a system is a tree, as usual, and moreover a wellfounded tree, as usual, but the tree can have infinite branching, so that the height of the tree may be transfinite. Such a proof is not a ‘real’ proof, that human beings communicate to each other, but intrinsically a metamathematical object.

The main tool in the study of upper bounds is cut elimination. We begin with a quick sketch of cut elimination for a classical ‘two sided’ sequent calculus.

5.1 Sequents, PA and PA$_\omega$

A *sequent* is an expression $\Gamma \Longrightarrow \Delta$ where Γ and Δ are finite lists of formulas $A_1, \dots A_m$ and $B_1, \dots, B_l$ respectively. One says that Γ *yields* Δ, or rather that the conjunction of the A's yields the disjunction of the B's. In particular, if both Γ and Δ are empty, the sequent asserts the impossible, i.e., a contradiction.

- Axioms

$$A \Longrightarrow A$$

- Cut

$$\frac{\Gamma \Longrightarrow \Delta, A \qquad A, \Lambda \Longrightarrow \Theta}{\Gamma, \Lambda \Longrightarrow \Delta, \Theta}\,\text{Cut}$$

  A is called the cut formula of the inference.

- Structural rules (exchange, weakening, contraction)

$$\frac{\Gamma, A, B, \Lambda \Longrightarrow \Delta}{\Gamma, B, A, \Lambda \Longrightarrow \Delta}\,\mathcal{X}_L \qquad \frac{\Gamma \Longrightarrow \Delta, A, B, \Lambda}{\Gamma \Longrightarrow \Delta, B, A, \Lambda}\,\mathcal{X}_R$$

$$\frac{\Gamma \Longrightarrow \Delta}{\Gamma, A \Longrightarrow \Delta}\,\mathcal{W}_L \qquad \frac{\Gamma \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta}\,\mathcal{W}_L$$

$$\frac{\Gamma, A, A \Longrightarrow \Delta}{\Gamma, A \Longrightarrow \Delta}\,\mathcal{C}_L \qquad \frac{\Gamma \Longrightarrow \Delta, A, A}{\Gamma, A \Longrightarrow \Delta, A}\,\mathcal{C}_R$$

- Negation

$$\frac{\Gamma \Longrightarrow \Delta}{\neg A, \Gamma \Longrightarrow \Delta}\,\neg_L \qquad \frac{B, \Gamma \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta, \neg B}\,\neg_R$$

- Implication

$$\frac{\Gamma \Longrightarrow \Delta, A \qquad B, \Lambda \Longrightarrow \Theta}{A \to B, \Gamma, \Lambda \Longrightarrow \Delta, \Theta}\,\to_L \qquad \frac{A, \Gamma \Longrightarrow \Delta, B}{\Gamma \Longrightarrow \Delta, A \to B}\,\to_R$$

- Conjunction

$$\frac{A, \Gamma \Longrightarrow \Delta}{A \wedge B, \Gamma \Longrightarrow \Delta}\,\wedge_{L1} \qquad \frac{\Gamma \Longrightarrow \Delta, A \qquad \Gamma \Longrightarrow \Delta, B}{\Gamma \Longrightarrow \Delta, A \wedge B}\,\wedge_{R}$$

$$\frac{B, \Gamma \Longrightarrow \Delta}{A \wedge B, \Gamma \Longrightarrow \Delta}\,\wedge_{L2}$$

- Disjunction

$$\frac{A, \Gamma \Longrightarrow \Delta \qquad B, \Gamma \Longrightarrow \Delta}{A \vee B, \Gamma \Longrightarrow \Delta}\,\vee_{L} \qquad \frac{\Gamma \Longrightarrow \Delta, A}{\Gamma \Longrightarrow \Delta, A \vee B}\,\vee_{R1}$$

$$\frac{\Gamma \Longrightarrow \Delta, B}{\Gamma \Longrightarrow \Delta, A \vee B}\,\vee_{R2}$$

- Quantifiers

$$\frac{F(t), \Gamma \Longrightarrow \Delta}{\forall x\, F(x), \Gamma \Longrightarrow \Delta}\,\forall_{L} \qquad \frac{\Gamma \Longrightarrow \Delta, F(a)}{\Gamma \Longrightarrow \Delta, \forall x\, F(x)}\,\forall_{R}$$

$$\frac{F(a), \Gamma \Longrightarrow \Delta}{\exists x\, F(x), \Gamma \Longrightarrow \Delta}\,\exists_{L} \qquad \frac{\Gamma \Longrightarrow \Delta, F(t)}{\Gamma \Longrightarrow \Delta, \exists x\, F(x)}\,\exists_{R}$$

Here a is a parameter (free variable), t is a term, and x is a bound variable.

In the intuitionistic version of the calculus, all sequents are required to have at most one disjunct. There are no rules corresponding to $\mathcal{C}_R$ or $\mathcal{X}_R$.

EXAMPLE 5 *A deduction of excluded middle*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \Longrightarrow A}{\Longrightarrow A, \neg A}\,\neg_R}{\Longrightarrow A, A \vee \neg A}\,\vee_{R2}}{\Longrightarrow A \vee \neg A, A}\,\mathcal{X}_R}{\Longrightarrow A \vee \neg A, A \vee \neg A}\,\vee_{\mathcal{R}\infty}}{\Longrightarrow A \vee \neg A, A \vee \neg A}\,\mathcal{C}_R$$

Cut Elimination (‘Hauptsatz’ = ‘fundamental theorem’): If a sequent is provable, it can be proved without cuts. As a corollary, if $\Gamma \Longrightarrow \Delta$ has a proof at all, it has a proof in which the only formulas that appear are subformulas of the formulas in Γ and Δ. As a corollary of that, the empty sequent has no proof.

EXAMPLE 6

$$\dfrac{\dfrac{A, \Gamma \Longrightarrow \Delta, B}{\Gamma \Longrightarrow \Delta, A \rightarrow B}\,\rightarrow_R \qquad \dfrac{\Lambda \Longrightarrow \Theta, A \qquad B.\Xi \Longrightarrow \Phi}{A \rightarrow B, \Lambda, \Xi \Longrightarrow \Theta, \Phi}\,\rightarrow_L}{\Gamma, \Delta, \Xi \Longrightarrow \Delta, \Theta, \Phi}\,\mathit{Cut}$$

can be transformed to

$$\dfrac{\dfrac{\Lambda \Longrightarrow \Theta, A \qquad A, \Gamma \Longrightarrow \Delta, B}{\Lambda, \Gamma \Longrightarrow \Theta, \Delta, B}\,Cut \qquad B, \Xi \Longrightarrow \Phi}{\Gamma, \Delta, \Xi \Longrightarrow \Delta, \Theta, \Phi}\,Cut$$

(There are still cuts, but the cut formulas are subformulas of the original cut.

The proof of cut elimination is rather intricate, as the process of transforming away instances of cut interferes with contraction. On account of this intricacy the height of a cut-free proof is (in the worst case) super-exponentially greater than the height of the original proof with cuts. Something like

$$4^{4^{4^{\cdot^{\cdot^{h}}}}}$$

where h is the height of the original proof, and the number of 4's is the length of the longest cut-formula that appears in it.

Not only does elimination of cuts make a proof (hugely) longer, it usually renders the proof also less *intelligible*. Cut formulas are rather like *lemmas*, and contain the idea of the proof.

So much for classical *logic*. A mathematical theory is usually based on axioms, and axioms tend to be rather poisonous for cut-elimination. (This is actually a little bit of a lie. The book [7] is an investigation of cut-elimination for several interesting mathematical theories, in cases where it *does* work. See the 'reasoned' bibliography at `http://www.helsinki.fi/~negri/ptpub.html`. Occasionally the axioms of a theory are of bounded complexity, and then one has partial cut-elimination, meaning that cuts can be eliminated which are of higher complexity than the axioms. This can pay off, for example when extracting bounds on the length of proofs of Π^0_2-statements[5] in systems with restricted induction schemes.)

One can avoid the problems with cut elimination by moving to an infinitary system. The so called ω-rule (due to Hilbert in 1931 and Schütte) consists of two kinds of infinitary inferences

$$\dfrac{\Gamma \Longrightarrow \Delta, F(0);\;\; \Gamma \Longrightarrow \Delta, F(1);\; \ldots\; \Gamma \Longrightarrow \Delta, F(n); \ldots}{\Gamma \Longrightarrow \Delta, \forall x\, F(x)}\,\omega_R$$

$$\dfrac{F(0), \Gamma \Longrightarrow \Delta;\;\; F(1), \Gamma \Longrightarrow \Delta;\; \ldots\; F(n), \Gamma \Longrightarrow \Delta; \ldots}{\exists x\, F(x), \Gamma \Longrightarrow \Delta}\,\omega_R$$

We make a version of PA, called PA$_\omega$ in which these rules replace $\forall_R$ and $\exists_L$. There are certain other (relatively minor) changes:

- We discard all free variables. All terms and formulas are closed. We use $\overline{n}$ as notation for the n-th numeral $S(S(\cdots S(0)\cdots))$ (in which there are n occurrences of the successor symbol S. All terms of PA$_\omega$ evaluate to a numeral.

- As axioms of PA$_\omega$ we take

[5] Formulas that begin $\forall n\, \exists m\, \ldots$.

- all sequents $\Longrightarrow A$ where A is a true atomic formula
- all sequents $B \Longrightarrow$ where B is a false atomic formula
- all sequents $F(s_1, \ldots, s_n) \Longrightarrow F(t_1, \ldots, t_n)$ where F is atomic, and for each $i = 1, \ldots, n$, s_i and t_i evaluate to the same numeral.

On this basis, we can now prove the (numerical) *induction axiom*

$$\Longrightarrow F(0) \wedge \forall x\, [F(x) \to F(S(x))] \to \forall x\, F(x)$$

As follows. First, for each n, there is a finite derivation $\mathcal{D}_n$ of the sequent $F(0), \forall x\, [F(x) \to F(S(x))] \Longrightarrow F(\overline{n})$. This can be proved by induction on n.

The basis is clear, so suppose we have $\mathcal{D}_n$. Let Δ be $F(0), \forall x\, [F(x) \to F(S(x))]$. Now construct $\mathcal{D}_{n+1}$ as follows.

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\mathcal{D}_n \\ \Delta \Longrightarrow F(\overline{n})\end{matrix} \qquad \begin{matrix}\text{structural rules} \\ F(\overline{n+1}), \Delta \Longrightarrow F(\overline{n+1})\end{matrix}}{F(\overline{n}) \to F(\overline{n+1}), \Delta \Longrightarrow F(\overline{n+1})}\,{\to_L}}{\forall x\, [F(x) \to F(S(x))], \Delta \Longrightarrow F(\overline{n+1})}\,{\forall_L}}{F(0), \forall x\, [F(x) \to F(S(x))] \Longrightarrow F(\overline{n+1})}\,\text{structural rules}$$

A final application of ω_R yields the desired proof of induction.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{matrix}\mathcal{D}_0 \\ \Delta \Longrightarrow F(\overline{0})\end{matrix}\ ;\ \cdots \quad \begin{matrix}\mathcal{D}_n \\ \Delta \Longrightarrow F(\overline{n})\end{matrix}\ ;\ \cdots}{F(0), \forall x\, [F(x) \to F(S(x))] \Longrightarrow \forall x\, F(x)}\,{\omega_R}}{F(0) \wedge \forall x\, [F(x) \to F(S(x))], \forall x\, [F(x) \to F(S(x))] \Longrightarrow \forall x\, F(x)}\,{\wedge_L}}{\forall x\, [F(x) \to F(S(x))], F(0) \wedge \forall x\, [F(x) \to F(S(x))] \Longrightarrow \forall x\, F(x)}\,{\mathcal{X}_L}}{F(0) \wedge \forall x\, [F(x) \to F(S(x))], F(0) \wedge \forall x\, [F(x) \to F(S(x))] \Longrightarrow \forall x\, F(x)}\,{\wedge_L}}{\dfrac{F(0) \wedge \forall x\, [F(x) \to F(S(x))] \Longrightarrow \forall x\, F(x)}{\Longrightarrow F(0) \wedge \forall x\, [F(x) \to F(S(x))] \to \forall x\, F(x)}\,{\to_R}}\,{\mathcal{C}_L}$$

We can now prove a cut-elimination theorem for PA$_\omega$. To state it we, need the notions of the height (an ordinal) and cut-rank (a natural number) of a derivation in PA$_\omega$. We write this

$$\mathcal{D} \vdash^{\alpha}_{n} (\Gamma \Longrightarrow \Delta)$$

and define this relation inductively, following the build-up of $\mathcal{D}$.

Define the length $|A|$ of a formula A as follows:

- $|A| = 0$ if A is atomic,
- $|\neg A| = |A| + 1$,
- $|A \wedge B| = \max(|A|.|B|) + 1$, and similarly for the other binary connectives $\vee$, $\to$,
- $|\exists x\, F(x)| = |\forall x\, F(x)| = |F(0) + 1$.

Suppose the last inference of $\mathcal{D}$ has the form

$$\cfrac{\begin{matrix}\mathcal{D}_0 & & \mathcal{D}_n & \\ \Gamma_0 \implies \Delta_0 \ ; \ \cdots & & \Gamma_0 \implies \Delta_0 \ ; \ \cdots\end{matrix}}{\Gamma \implies \Delta}\ \mathcal{I}$$

where the number of premises is either 0, 1, 2, or ω, and the $\mathcal{D}_n$ are the immediate subderivations of $\mathcal{D}$.

If

$$\mathcal{D}_n \vdash_k^{\alpha_n} (\Gamma_n \implies \Delta_n)$$

and for each n, $\alpha_n < \alpha$, then

$$\mathcal{D} \vdash_k^{\alpha} (\Gamma \implies \Delta)$$

provided that in the case where $\mathcal{I}$ is a cut, with cut formula A, then also $|A| < k$.

We now state a number of facts.

- The first concerns the embedding of ordinary first order PA into PA_ω. If $\Gamma \implies \Delta$ is provable in PA, then there is a proof $\mathcal{D}$ and natural numbers m, n such that
$$\mathcal{D} \vdash_n^{\omega+m} \Gamma \implies \Delta$$
- The second lemma contains the heart of cut-elimination. Suppressing explicit mention of derivations, If $\vdash_k^{\alpha} \Gamma \implies \Delta, A$ where $|A| = k$, and $\vdash_k^{\beta} A, \Lambda \implies \Theta$, then $\vdash_k^{\alpha\#\beta} \Gamma, \Lambda \implies \Delta, \Theta$. ($\alpha\#\beta$ is the Hessenberg natural sum of α and β, mentioned in the section on arithmetic. See also http://en.wikipedia.org/wiki/Ordinal_arithmetic.)
- The third lemma builds upon the last lemma. By eliminating all cuts of the highest rank in a a derivation, we arrive at a longer derivation. The lemma states an upper bound (which happens to be best-possible) on the height of the new derivation.

  If $\vdash_{k+1}^{\alpha} (\Gamma \implies \Delta)$ then $\vdash_k^{\omega^\alpha} (\Gamma \implies \Delta)$.

  (Having said that, I am in some doubt whether instead of ω^α above I should have said 2^α. If any one is reading this, what do you think?)
- Now we eliminate *all* cuts.

  If $\vdash_n^{\alpha} (\Gamma \implies \Delta)$ then $\vdash_0^{t(n,\alpha)} (\Gamma \implies \Delta)$, where
$$t(n,\alpha) = \omega^{\omega^{\omega^{\cdot^{\cdot^{\alpha}}}}}$$
  and there are n ω's in the tower of exponents.

By further elaboration of this proof, one can wring out of it upper-bounds on the extent to which transfinite induction can be proved in PA.

You may think this proof is less than breathtakingly elegant, and to a certain extent I agree. However, in this aesthetic vein, the idea of avoiding the problem of bad interaction between mathematical axioms and cut elimination by use of infinitary rules is bold and incisive, if drastic. Aesthetics aside, it has to be admitted that it is still a little mysterious exactly why the move to infinitary rules works. It should be mentioned that Wilfried Buchholz has shed some light on the question, in papers from 1991 and 1997. The key idea is that can regard finite derivations (of the ordinary, humanly communicable kind) as *notations* for infinitary derivations. The titles of the relevant papers are "Notation Systems for Infinitary Derivations" and "Explaining Gentzen's Consistency Proof within Infinitary Proof Theory".

Quite a bit of the clutter in the proof is superficial. One can improve the presentation by using what is called a 'one-sided' or 'Tait-style' sequent calculus. In such a calculus (which is intrinsically classical), the negation connective is thrown out, and replaced by a defined operation that relies on each atomic formula splitting into a positive and a negative form . Also, implication is thrown out, so that the only logical operations that remain are conjunction and disjunction, and the universal and existential quantifiers. Or we could say that disjunction and conjunction come in both a binary and a countable form.

In recent proof theory of stronger systems than arithmetic, infinitary systems based on Kripke-Platek set theory often play the role that is played here by the system PA_ω. The subject is (regrettably) extremely technical. But perhaps enough time has passed that one can begin to look for a revision of the elementary material in a form which is more amenable to an algebraic or categorical description.

5.2 Natural Deduction: infinitary types and terms

We now turn from infinitary sequent calculus to infinitary lambda-calculus (i.e., natural deduction), in this following Tait [8] and (particularly) Martin-Löf [9]. One novelty here is that the *types* (formulas) as well as the terms (derivations) are infinitary.

Part of my reason for presenting these ideas is that they are in some respects simpler than the sequent calculus counterparts, and, undeservedly, not (particularly) well known.

Types The types $\sigma, \tau, \ldots$ of this system contain at least one atomic type, and are closed under the (non-dependent) function spaces $\sigma - < \tau$, and countable conjunction: if $\tau_0, \tau_1, \ldots, \tau_n, \ldots$ is a countable sequence (stream) of types, then

$$\Pi_n \tau_n$$

is a type. This is the type of functions whose domain is $\mathbb{N}$ and whose value for argument n has type τ_n. To use stream terminology, it is a type of *heterogeneous streams* $t_0, t_1, \ldots, t_n, \ldots$ such that

$$\begin{array}{l} t_0 : \tau_0 \\ (t_1, t_2, \ldots, t_{n+1}, \ldots) : \Pi(\tau_1, \tau_2, \ldots, \tau_{n+1}, \ldots) \end{array}$$

Terms The terms of this calculus, so far as function types are concerned are built up using λ-abstraction and application in the usual way.

As for the terms Π-types, the relevant term forms are these. First an *infinitary operation*

$$\frac{t_0 : \tau_0;\;\; t_1 : \tau_1;\;\; \ldots t_n : \tau_n;\;\; \ldots}{(t_0, t_1, \ldots, t_n, \ldots) : \Pi(\tau_0, \tau_1, \ldots, \tau_n, \ldots)}$$

Second, countably many *projection* operations, one for each natural number n. Martin-Löf writes it as follows

EXAMPLE 7 *Using variables of type τ_0 and $\Pi_n(\tau_n \to \tau_{n+1})$ respectively, then*

$$\lambda\,x\,\lambda\,y\,(x, y\,0\,x.y\,1(y\,0\,x), \ldots)$$

is an example of a term of type $\tau_0 \to \Pi_n(\tau_n \to \tau_{n+1}) \to \Pi_n\tau_n$

Computations The reduction rules of the infinitary calculus are based on the following two *contraction* rules. First, β-contraction

$$(\lambda\,x\,t(x))\,s \rightsquigarrow t(s)$$

(where $t(s)$ denotes the result of substituting s for all free occurrences of the variable x in the term $t(x)$, renaming any troublesome bound variables in $t(x)$ as necessary to avoid capturing free occurrences of variables in s. The use of parentheses is heavily overloaded here.)

Second, projection

$$(t_0, t_1, \ldots, t_n, \ldots)\,n \rightsquigarrow t_n$$

The *reduction* relation is formulated as follows. It is the relation inductively generated by the following clauses.

- A variable reduces to itself.
- If $s(x)$ reduces to $t(x)$, then $\lambda\,x\,s(x)$ reduces to $\lambda\,x\,t(x)$.
- If r reduces to s, then $r\,t$ reduces to $s\,t$.
- If r reduces to s, then $t\,r$ reduces to $t\,s$.
- If s_n reduces to t_n for $n = 0, 1, \cdots$, then $(s_0, s_1, \ldots)$ reduces to $(t_0, t_1, \ldots)$.
- If r reduces to s, then $r\,n$ reduces to $s\,n$.
- If r reduces to s and s reduces to t, then r reduces to t.

EXERCISE 9 *Does this calculus have any infinite reduction sequences?*

EXERCISE 10 *In a certain sense, combinators* $S : (\tau \to \sigma \to \rho) \to (\tau \to \sigma) \to \tau \to \rho)$ *(with* $S\,r\,s\,t \leadsto r\,t\,(s\,t)$*) and* $K : \sigma \to \tau \to \sigma$ *(with* $K\,s\,t \leadsto s$*) are sufficient to allow for* λ*-abstraction to be simulated in the simply-types* λ*-calculus.*

Can you find a (countable) set of combinators (with associated contractions) that would allow (in some reasonable sense) λ*-abstraction to be simulated.*

A passage from Martin-Löf's paper:

> We are now prepared to establish the isomorphism between the system of terms and this system of natural deduction. The following dictionary shows the relation.
>
> | | |
> |---|---|
> | atomic type | atomic formula |
> | type | formula |
> | variable | assumption |
> | bound variable | discharged assumption |
> | rule of term formation | deduction rule |
> | term | deduction |
> | λ-contraction | $-$-reduction |
> | projection | *logand*-reduction |
> | normal term | cut free deduction |
>
> Curry and Feys 1958 discovered the analogy between their so called theory of functionality and the positive implicational calculus, and Howard 1969 extended it to Heyting arithmetic. I am indebted to William Howard for pointing out this analogy to me.

EXERCISE 11 *Think about what the various types, terms, and contraction rules mean according to the Curry-Howard correspondence.*

Assuming you've had a go at the previous exercise, what are the axioms *that correspond to the combinators you came up with?*

Ordinals The *degree* $d(\tau)$ of a type τ (which corresponds approximately to the length of a formula in the sequent calculus treatment) is defined as follows.

- The degree of an atomic type is 0.
- $d(\sigma \to \tau) = \max((d(\sigma) + 1, d(\tau))$.
- $d(\Pi_n\, \tau_n) = \max_n d(\tau_n)$.

The *cut type* of a redex of the form $(\lambda\, x\, t(x))\, s$ or $(t_0, t_1, \ldots)\, n$, is the type of $(\lambda\, x\, t(x))$ or $(t_0, t_1, \ldots)$, respectively. The *cut degree* of a term is the maximum of the degrees of all its cut types.

The *length* $l(t)$ of a term t is defined as follows.

- The length of a variable is 0.

- $l(\lambda\, x\, t(x)) = l(t(x)) + 1$
- $l(t\, s) = \max(l(s) + 1, l(t))$
- $l(t_0, t_1, \ldots)) = \max_n(l(t_n))$

EXAMPLE 8 *The length of the recursion operator* $\lambda\, x\, \lambda\, y\, (x, y\, 0\, x.y\, 1(y\, 0\, x), \ldots)$ *in* $\omega + 2$.

By induction on the structure of $t(x)$ one can prove

$$l(t(s)) \leq l(s) + l(t(x))$$

The aim now is to prove not only that each term reduces to a normal form (containing no redex), but to estimate the length of this term in terms of the length and cut degree of the starting term. Martin-Löf now carefully defines a certain binary function $\phi_\alpha \beta$ that is based on (but not the same as) the Veblen hierarchy over the normal function 2^α. Let us write this hierarchy (binary function) $\chi_\beta \alpha$. So $\chi_0 \alpha = 2^\alpha$, and for $\beta > 0$, χ_β enumerates the common fixed points of all χ_γ for $\gamma < \beta$. Let χ_β^m denote the m-th (finite) iteration of the function χ_β.

The function ϕ is now defined by $\phi_0(\alpha) = \alpha$, and

$$\phi_\beta(\alpha) = \chi_{\beta_1}^{m_1}(\chi_{\beta_2}^{m_2}(\cdots \chi_{\beta_k}^{m_k}(\alpha) \cdots))$$

where

$$\beta = \omega^{\beta_1} m_1 + \omega^{\beta_2} m_2 + \cdots + \omega^{\beta_k} m_k$$

is the Cantor normal form of $\beta > 0$ to the base ω.

If you have any experience of working in ordinal analysis, either in upper bounds or lower bounds, you encounter this function again and again. Note: it is not so much the Veblen hierarchy itself that plays a role in ordinal analysis, but the function based on it as ϕ is to χ.

The function ϕ enjoys the following rather pretty property. It is a solution to the equation

$$\phi_\beta \cdot \phi_\gamma = \phi_{\beta+\gamma}$$

under the initial conditions $\phi_0(\alpha) = \alpha$ and $\phi_1(\alpha) = 2^\alpha$.

EXERCISE 12 *Try to prove this claim.*

There are three properties of the functions ϕ_β used in the proof of the normalisation theorem.

- ϕ_β is strictly increasing for all β.
- $\phi_\beta \cdot \phi_\gamma = \phi_{\beta+\gamma}$
- $\phi_\beta(\alpha) \times 2 \leq \phi_\beta(\alpha + 1)$. It is to obtain this property for $\beta = 1$ that we chose $\phi_1(\alpha) = \chi_0(\alpha) = 2^\alpha$.

So much for the ordinal arithmetic involved. Note though that it is not at all straightforward to develop the notions of degree, cut degree and length directly for the Brouwer ordinals. This is because of the use of max in the definitions of these notions, which is not constructively definable. (The order relation between Brouwer ordinals is not decidable.) Instead, we have to work with an ordinal representation system, together with a total order on the terms.

Normalisation We are heading for the following result

PROP 3 *A term of length α and cut degree β reduces to a normal term of length $\leq \phi_\beta(\alpha)$.*

Notice first that when we eliminate a cut type from a term t by contracting a subterm, the cut degree of the new term t' is not increased. This is because the degrees of the new cut types that may be introduced do not exceed the cut type which is eliminated.

Furthermore, all cut terms of the form $\Pi_n \tau_n$ can be eliminated from a term t without increasing its length. Prove this by induction on the structure of t. (It should be fairly obvious.)

Now we come to the heart of the proof, which is concentrated in the following lemma.

LEMMA 5 *A term r of length α and cut degree $\beta + \gamma$ reduces to a term of length $\leq \phi_\gamma(\alpha)$ and cut degree $\leq \beta$.*

This proved by (order) induction on γ, and within that by (structural) induction on r.

The proof is quite intricate – it would not be an attractive task to formalise and check it using a proof checker. To illustrate this, I reproduce here part of the proof.

Remember that the outermost induction is transfinite induction on γ. By the preliminary lemma we can assume that r has no cut type of the form $\Pi_n \, \tau_n$. The inner induction is a structural induction on r, in which the basis case is that of a variable. For the induction step there are four cases to distinguish, depending on the form of r: $\lambda \, x \, t(x)$, $(t_0, t_1, \ldots)$, $t \, s$, and $t \, n$. Here is the third case, that of an expression of applicative form.

By (inner) induction hypotheses, s and t reduce to some expressions u and v where $l(u) \leq \phi_\gamma(l(s))$ and $l(v) \leq \phi_\gamma(l(t))$ and the cut degrees of u and v are both $\leq \beta$. If v is *not* of abstraction form, we are done, because then r reduces to $v \, u$ which has length at most

$$\max(\phi_\gamma(l(s)) + 1, \phi_\gamma(l(t)) \leq \phi(\max(l(s) + 1, l(t))) = \phi_\gamma(\alpha)$$

and cut degree $\leq \beta$. Otherwise, r must have the form

$$\lambda \, x \, (t(x) \, s_1 \cdots s_n)$$

where $s_n = s$ and each s_i is either a term or a natural number. Let the maximum of β and the degrees of the types of the s_i that are terms be $\beta + \delta$. Then $\delta < \gamma$

and $\gamma - \delta \leq \gamma$. By (outermost) induction hypothesis, $t(x)$ reduces to $v(x)$, and this has length $\leq \phi_{\gamma-\delta}(l(t(x)))$ and cut degree $\leq \beta+\delta$. Also, if s_i is a term, then s_i reduces to some u_i which has length $\leq \phi_{\gamma-\delta}(l(s_i))$ and cut degree $\leq \beta + \delta$. If s_i is a natural number, put $u_i = s_i$. Then r reduces to

$$\lambda\, x\, (v(x)\, u_1 \cdots u_n)$$

and at most n contractions reduce this term to a term w of length at most

$$\begin{array}{ll} & \max_i l(u_i) + l(v(x)) \\ \leq & \phi_{\gamma-\delta}(\max_i l(s_i)) + \phi_{\gamma-\delta}(l(t(x))) \\ \leq & \phi_{\gamma-\delta}(\max(l(t(x)), \max_i l(s_i))) \times 2 \\ \leq & \phi_{\gamma-\delta}(\max(l(t(x)), \max_i l(s_i)) + 1) \\ = & \phi_{\gamma-\delta}(\alpha) \end{array}$$

and cut degree $\leq \beta+\delta$. Finally, w reduces to a term of length $\leq \phi_\delta(\phi_{\gamma-\delta}(\alpha)) = \phi_\gamma(\alpha)$ and cut degree $\leq \beta$.

6 TODO

- Something about notation system as coalgebras. Examples for ϵ_0 and Γ_0.
- structural induction/recursion versus order-induction.

References

[1] Saunders Mac Lane. Despite physicists, proof is essential in mathematics. *Synthese*, 111(2):147–154, May 1997.

[2] David Hilbert. Über das unendliche. In Jean van Heijenoort, editor, *From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1831*, pages 367–392. Harvard University Press, Cambridge, Massachusetts, 1967. (Text of talk given in 1925. First published in Mathematishe Annalen 95, 1926).

[3] Anne S. Troelstra and Helmut Schwichtenberg. *Basic Proof Theory*, volume 43 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1996. [A solid introduction to the more basic parts of structural proof theory, covering natural deduction systems, hilbert systems, sequent calculi, the modal logic S4, and linear logic. Contains a chapter on instances of transfinite induction provable in first order arithmetic. It is probably worth searching out the 2nd edition, it which many small errors have been fixed. See also Schwichtenberg's publications page at `http://www.mathematik.uni-muenchen.de/~schwicht/publikationen.html` for corrections to the 2nd edition.].

[4] M. Rathjen. The realm of ordinal analysis. In *Proceedings of the Logic Colloquium '97, Cambridge*. Cambridge University Press, 1997. [A survey

paper. Rathjen is a leading researcher in modern proof theory, and writes very well for a general mathematical audience. Thoroughly recommended.].

[5] Richard L. Epstein and Walter A. Carnielli. *Computability (2nd ed.): computable functions, logic, and the foundations of mathematics.* Wadsworth Publ. Co., Belmont, CA, USA, 2000.

[6] Panu Raatikainen. Hilbert's program revisited. *Synthese*, 137(1 - 2):157–177, November 2003. [Carefully teases apart the issues involved in assessing the impact of Gödel's incompleteness theorems on Hilbert's program.].

[7] Sara Negri and Jan von Plato. *Structural Proof Theory.* Cambridge University Press, Cambridge, 2001. [I have been told that this book is an excellent introduction to structural proof theory, that also contains interesting new results about cut elimination in the presence of mathematical axioms of a particular form. Also connected is an interactive editor ('Pesca': `http://www.cs.chalmers.se/~aarne/pesca/`) for sequent calculus proofs, written by Aarne Ranta].

[8] W.W. Tait. Infinitely long terms of transfinite type. In Dummett M.A.E. Crossley, J.N., editor, *Formal Systems and Recursive Functions*, Studies in Logic and the Foundations of Mathematics, pages 176–185, Amsterdam, 1965. North-Holland. [An early paper, inspired in part by an unpublished extension by William Howard of Gödel's 'Dialectica' interpretation to ramified analysis, using transfinite types, and in part by the use by Novikov (1943), Lorenzen (1951) and Scütte (1954)of infinite well-founded proof trees.].

[9] P. Martin-Löf. Infinite terms and a system of natural deduction. *Compositio Mathematica*, 24:93–103, 1972. [A simplified formulation of Tait's system that more fully expresses the relation to infinitary proof theory that is implicit in Tait's paper.].

[10] W. Pohlers. *Proof Theory, an Introduction*, volume 1407 of *Lecture Notes in Mathematics.* Springer-Verlag, Berlin, Heidelberg, New York, 1989. [Perhaps the best published introduction now on ordinal-theoretic proof theory, covering a lot of ground I don't even hint. In particular, it deals with the formal theory of non-iterated inductive definitions, and the use of collapsing functions in devising ordinal representation systems.].

[11] K. Schütte. *Proof Theory.* Springer, Berlin, Heidelberg, New York, 1977. [Very dry. A revision of a book published in German in 1960 by a major figure in the subject.].

[12] Gaisi Takeuti. *Proof Theory.* Elsevier Science Publishers, 2 edition, 1987. [I have not read this myself, but it is often recommended. I think there are (in the 2nd edition) appendices by Kreisel, Feferman, Simpson and Pohlers.].

[13] J. Y. Girard. *Proof Theory and Logical Complexity.* Bibliopolis, 1987. [Rather difficult to find. I have seen only parts of it: a particularly good account of Herbrand's theorem, and functional interpretations (as I recall). Almost anything Girard writes is worth reading, if only for his mercurial style and iconoclasm.].

[14] Gerhard Gentzen. *The Collected Papers of Gerhard Gentzen.* North Holland, 1969. Edited by M. E. Szabo. [Nuff said.].

[15] Stephen G. Simpson. *Subsystems of second order arithmetic.* Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1999. [A subject called 'reverse mathematics' has two main proponents, Steve Simpson and Harvey Friedman. It seeks to discover which logical principles, expressed in fragments of second order arithmetic are necessary to prove theorems of mathematics – like Heine-Borel, Hahn-Banch, Bolzano-Weierstrass, Cantor-Bendixon. (They don't have to be double-barrelled!). A lively, and peculiarly American branch of proof theory. I know very little about the subject, and have never even seen the book. But it should certainly be mentioned. I recommend reading the Wikipedia entry. A second edition is due to be published Real Soon Now.].

[16] Jean Gallier. What's so special about kruskal's theorem and the ordinal [gamma]o? a survey of some results in proof theory. *Annals of Pure and Applied Logic*, 53(3):199–260, September 1991. [Abstract: This paper consists primarily of a survey of results of Harvey Friedman about some proof-theoretic aspects of various forms of Kruskal's tree theorem, and in particular the connection with the ordinal [Gamma]0. We also include a fairly extensive treatment of normal functions on the countable ordinals, and we give a glimpse of Verlen hierarchies, some subsystems of second-order logic, slow-growing and fast-growing hierarchies including Girard's result, and Goodstein sequences. The central theme of this paper is a powerful theorem due to Kruskal, the 'tree theorem', as well as a 'finite miniaturization' of Kruskal's theorem due to Harvey Friedman. These versions of Kruskal's theorem are remarkable from a proof-theoretic point of view because they are not provable in relatively strong logical systems. They are examples of so-called 'natural independence phenomena', which are considered by most logicians as more natural than the metamathematical incompleteness results first discovered by Godel. Kruskal's tree theorem also plays a fundamental role in computer science, because it is one of the main tools for showing that certain orderings on trees are well founded. These orderings play a crucial role in proving the termination of systems of rewrite rules and the correctness of Knuth-Bendix completion procedures. There is also a close connection between a certain infinite countable ordinal called [Gamma]o and Kruskal's theorem. Previous definitions of the function involved in this connection are known to be incorrect, in that, the function is not monotonic. We offer a repaired definition of this function, and explore briefly the consequences of its existence.].

[17] Jeremy Avigad and 2001 Erich H. Reck, Carnegie Mellon Technical Report CMU-PHIL-120. Clarifying the nature of the infinite: the development of metamathematics and proof theory. Technical Report CMU-PHIL-120, Carnegie Mellon, 2001. [A well-written survey article, that focuses particularly on the origins of proof theory. An extensive bibliography.].

Some URL's

The following is a partial list of homepages for people who work or have worked in ordinal theoretic proof theory. One can get an impression of what is going on in this subject by browsing among their online publications.

- Jeremy Avigad `http://www.andrew.cmu.edu/~avigad/`
- Wilfried Buchholz `http://www.mathematik.uni-muenchen.de/~buchholz/`
- Michael Rathjen `http://www.amsta.leeds.ac.uk/Pure/staff/rathjen/`
- Toshiyasu Arai `http://kurt.scitec.kobe-u.ac.jp/~arai`
- Gerhard Jäger `http://www.iam.unibe.ch/~til/staff/jaeger.html`
- Andreas Weiermann `http://cage.rug.ac.be/~weierman/`
- Anton Setzer `http://www.cs.swan.ac.uk/~csetzer/`
- Lev Beklemishev `http://www.phil.uu.nl/~lev/`
- Sol Feferman `http://math.stanford.edu/~feferman/`
- Stan Wainer `http://www.amsta.leeds.ac.uk/Pure/staff/wainer/wainer.html`
- Bill Tait `http://home.uchicago.edu/~wwtx/`
- Thomas Strahm `http://www.iam.unibe.ch/~strahm`
- Sergei Tupailo `http://www.cs.ioc.ee/~sergei/`
- Harvey Friedman `http://www.math.ohio-state.edu/~friedman/`
- Steve Simpson `http://www.math.psu.edu/simpson/`
- Tim Carlson `http://www.math.ohio-state.edu/~carlson/`
- Andrea Cantini `http://www.philos.unifi.it/CMpro-v-p-160.html`
- `http://www.prooftheory.org/`, `http://en.wikipedia.org/wiki/Proof_theory`, ...

I have almost certainly forgotten one or more crucial names.