

Iteration templates as generalized ordinal notations

Harold Simmons

School of Mathematics, The University, Manchester, England
hsimmons@manchester.ac.uk

Abstract

This document is a survey of the four papers [λH], [Sch], [Com], [FandH]. It gives the relevant background and shows how the papers fit together to form a whole. Material from those papers is repeated only when necessary.

As it stands the document is not intended for publication. However, if there is sufficient interest then I may rewrite it to include [λH], [Sch], [Com], [FandH] and so form a self contained development.

This material has been around and available in some form for quite a while. This version was slightly modified after attending the Leeds meeting on Proof Theory in July 2009. I have added one or two extra comments in the light of my current knowledge.

Contents

1	Introduction	1
2	The systems λG and λH	10
3	The semantics of λG and λH	16
4	Limit lifting	21
5	The category of limit structures	24
6	A system of informal notations	32
7	Naming in λG	36
8	Equational reasoning and canonical notations	39
9	The arithmetic of iterators in λH	43
10	Naming in λH	51
11	Fruitful and helpful ordinal functions	56
12	Interpretation in $\mathbb{O}rd$	65
13	Veblen hierarchies	74
14	Uncountable ordinals as templates	81
15	The curly gadgets	87
16	An enumerating function	96
17	Present Development	101
	References	109

1 Introduction

As indicated in the Abstract this survey gives the background material to the three papers [λH], [Sch], [Com], [FandH]. It is not intended for publication (in its present form) and so can proceed at a more leisurely pace. On the whole I will not repeat material from those papers, but I will indicate the parts relevant to a current discussion. Nevertheless, there are a few places in the survey where it is convenient to have some repetition.

What is this survey about? As a first approximation I could say ‘ordinal notations’ but that would miss a crucial aspect. We are concerned here with nested iterations

and how such constructs can be described. How can we format a set of instructions for performing such iterations? Of course, ordinal notations are the best known examples of such descriptions, but they are not the whole story. It is the nestings within the constructs and not the well founded aspects that are the main concern.

Consider any concrete set \mathbb{A} and let

$$\mathbb{A}' = (\mathbb{A} \rightarrow \mathbb{A})$$

be the set of functions on \mathbb{A} . Given any such function $f : \mathbb{A}'$ we form the finite iterates f^m (for $m \in \mathbb{N}$) in the obvious way. Thus

$$f^0 a = a \quad f^{m+1} a = f(f^m a)$$

for each $a \in \mathbb{A}$ and $m \in \mathbb{N}$. Starting from any element $a \in \mathbb{A}$ these iterates

$$(f^r a \mid r < \omega)$$

trace out an ω -chain through \mathbb{A} . Given some extra information we may extend this chain. We first somehow construct an element we may call $f^\omega a$ and then continue to use f . After that there is, within reason, no stopping the process.

But, you may complain, what I have just described is an ordinal iteration. Not quite, for if we delve a bit deeper we find that the behaviour at a limit ordinal depends on the notation for that ordinal rather than just the ordinal value. We will look at this shortly. There is also something else that is hidden by the informal description.

Certainly when any iteration is performed it will produce an ordinal indexed chain of elements (in the ambient set \mathbb{A}). However, the way that this chain is generated may depend on more than just the ordinal. It can depend on the notations used for the ordinals involved, and perhaps even the syntax used to describe those.

To get some idea of the difference between ordinal and ordinal notation let's look at a couple of examples. Further examples are given Section 2 of [λH], and other examples will be used later in this survey. After these particular examples I will give a rough idea of the more general notion of an 'iteration template'. (At this stage it can only be a rough idea, but it will become sharper as the survey develops.)

1.1 EXAMPLE. (Ordinal) Consider the Cantor-Bendixson process on a topological space S . Let D be the operation which converts a closed subset X of S into its subset $DX \subseteq X$ of limit points. The set DX is also closed, and so we can iterate D at least finitely many times. Also, by doing something appropriate at limit levels we can iterate along the ordinals. Thus we set

$$D^0 X = X \quad D^{\alpha+1} = D(D^\alpha X) \quad D^\lambda X = \bigcap \{D^\alpha X \mid \alpha < \lambda\}$$

for each closed set X , ordinal α , and limit ordinal λ . This generates an ordinal indexed family $D^\alpha X$ of closed subsets of X . This is the modern version of the example which lead Cantor to invent (or discover) the ordinals.

This process can be continued all the way along the ordinals. Of course, for any particular space S the process will eventually stabilize (at or before the Cantor-Bendixson rank of S), but an arbitrarily large ordinal can be achieved by a suitably chosen space.

I have been told that Cantor did not think of ordinals as we now view them, but in terms of iteration. The notion of an ordinal condensed out of this idea. Here I am suggesting that we should not forget this original idea.

(Ordinal notation) For a second example consider one of the many hierarchies of number theoretic functions that appear in some parts of proof theory. We start from some suitable function $f : \mathbb{N}'$ and generate a hierarchy by

$$f_0 = f \quad f_{\alpha+1}u = f^{u+1}u \quad f_\lambda u = f_{\lambda[u]}u$$

for each $u \in \mathbb{N}$, each ordinal α , and each limit ordinal λ *with a selected fundamental sequence* $\lambda[\cdot]$.

The result of this process is often described as an ordinal-indexed family of functions. That, of course, is just a convenient way of speaking. As emphasized, the result at a limit ordinal λ and beyond depends of the fundamental sequence selected for λ . Usually we have in mind a uniform way of selecting these, and most of the standard selections give results with approximately the same complexity. However, it is possible to alter quite dramatically the generated functions merely by changing the selected fundamental sequences.

There is also another difference between this and the previous example. If we want a fundamental sequence to be an ω -sequence, then we can never get out of the countable. More importantly, if we wish to use some reasonably uniform selection of fundamental sequences, then there will be an upper bound on the use of this method. In practice this doesn't matter for there is often a level beyond which we don't want to go (as yet). ■

What should we make of this second example? There are three observations.

- There will always be a cofinality problem which restricts how far we can go. Let's here and now decide that, for the moment, we are quite happy to stick with the countable.
- In general a bare limit ordinal will not be enough. We need to know which of its fundamental sequences we are suppose to use. One way to deal with this is to think of some system of dressed ordinals in which each ordinal comes clothed with all the fundamental sequences used in its construction. Ordinals can be dressed in many different ways. This gives a much larger and richer system of gadgets over which we may iterate. In particular, the system is certainly not linearly ordered. We can then view iterations such as Example 1.1(b) as taking some path through this system.
- Once we have made the step to dressed ordinals, there is another question we should consider. The most characteristic feature of an ordinal is that it is well ordered. But what has the more general kind of iteration got to do with well foundedness?

Think of the two examples in a more general context. We have some domain \mathbb{A} of entities. In the first \mathbb{A} is the set of closed subsets of a space, in the second it is the family of functions in \mathbb{N}' . We have a process

$$A : \mathbb{A} \longrightarrow \mathbb{A}$$

of converting one entity into another. Thus in the two examples we have

$$A = D \quad A = \mathbf{ack}$$

where

$$\mathbf{ack}fu = f^{u+1}u$$

for each $f : \mathbb{N}'$, $u \in \mathbb{N}$. We think of this process A as a generalized successor function. Starting from some given $a \in \mathbb{A}$ we repeatedly use A to generate the finite iterates $A^u a$ of A on a (where $u \in \mathbb{N}$). This produces an ω -sequence $p : \mathbb{N} \longrightarrow \mathbb{A}$ through \mathbb{A} where $pu = A^u a$. We wish to continue this process, so somehow we must collate the values of p into a single member of \mathbb{A} , and then we can start again. In short, we need a higher order function

$$\mathcal{A} : (\mathbb{N} \longrightarrow \mathbb{A}) \longrightarrow \mathbb{A}$$

which converts ω -sequences into single values. In many natural examples this \mathcal{A} is some kind of ‘limit creator’, but there is no reason why it can’t be any function of the indicated type (unless we choose to loosen our idea of limit).

In the general situation we have a structure

$$\mathfrak{A} = (\mathbb{A}, a, A, \mathcal{A})$$

which we think of as providing the data with which we may perform ‘long’ iterations. Consider how such an iteration could be described. The temptation is to think in terms of ordinal iterations. Starting from a repeatedly use A and every so often use \mathcal{A} to collate into a new starting element. The template for such an iteration would be a dressed ordinal. However, other forms of iteration are possible. What about the following kind of instruction?

The element we want is $\mathcal{A}p$ where $p : \mathbb{N} \longrightarrow \mathbb{A}$ is a certain function. For each $u \in \mathbb{N}$ the element pu is obtained by a certain iteration. For instance, we may have

$$pu = A(\mathcal{A}q_u)$$

(for $u, v \in \mathbb{N}$) where, for each $u \in \mathbb{N}$ the function $q_u : \mathbb{N} \longrightarrow \mathbb{A}$ is itself obtained by a certain iteration. Thus we could set

$$q_u v = A^{f(u,v)}(\mathcal{A}r_{u,v})$$

where

$$f : \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N}$$

is some function and for each $u, v \in \mathbb{N}$ the function $r_{u,v} : \mathbb{N} \longrightarrow \mathbb{A}$ is obtained in a certain way, perhaps by iteration. These in turn may bring in deeper iterations, and so on.

What we have here is an informal template which, given any structure \mathfrak{A} will trace out a path through the carrier \mathbb{A} . The template consists of a highly nested collection of instructions for performing smaller, less nested, iterations and combining these in a certain way. This template may refer to some external gadgets, such as the function f in the example. How can we make this idea precise?

1.2 THE IDEA. An iteration template is a finite piece of syntax which, for each structure \mathfrak{A} , as above, determines an element of the carrier. This element is located by unravelling the syntax of the template into smaller templates, and so on recursively. ■

This survey and the papers [λH], [Sch], [Com], [FandH] are an attempt to make this idea precise and to work out some of the consequences. Of course, an ordinal notation (in the sense of a dressed ordinal) will be an iteration template, and these will form an

important stock of examples. However, there is no reason to think that every iteration template is ‘equivalent’ to some standard ordinal notation.

The phrase ‘iteration template’ is a bit of a mouthful. At some point we ought to find a slicker name, but for the time being we will chew on what we’ve got.

To make the Idea 1.2 precise there are two questions we must think about.

(Syntax) How should we set up the syntactic mechanisms which organize the iteration templates? This can not be a low level mechanism, for as well as the iteration templates themselves we have to organize various higher level gadgets (such as the sequences p which are collated).

(Semantics) Is there some concrete set \mathbb{O} which we can think of as the collection of all concrete iteration gadgets. It may be that two iteration templates (as pieces of syntax) always produce the same result no matter which structure \mathfrak{A} they operate on. This puts some kind of equivalence relation on templates, and we want the elements of \mathbb{O} to represent the blocks of this equivalence.

As often happens, neither of these questions can be answered before the other. Each partial answer to one question has an impact on any previous partial answer to the other and so improves that earlier answer which in turns leads to a better answer than the one we have just thought of. Let me simply describe what will be developed here without attempting to explain how I came to this position.

The syntactic mechanism which organizes iteration templates and the associated gadgetry is an applied λ -calculus $\lambda\mathbf{H}$. It has an atom (ground type) \mathcal{O} which is thought of as the name of the concrete set \mathbb{O} , even though we don’t yet know what \mathbb{O} is. The hope is that an analysis of $\lambda\mathbf{H}$ should give us a better idea of this hazy set. As usual the system $\lambda\mathbf{H}$ will produce judgements

$$\Gamma \vdash t : \tau$$

well-formed terms in context. In particular, a term

$$\vdash \alpha : \mathcal{O}$$

in the empty context will be an iteration template. Of course, to produce α we may need to pass through several other levels of syntax, and the system $\lambda\mathbf{H}$ must cater for these.

We design the system $\lambda\mathbf{H}$ as an extension of a more familiar system $\lambda\mathbf{G}$ which is concerned only with numeric gadgets.

The system $\lambda\mathbf{G}$ is the term calculus of Gödel’s T . This can be thought of as a systematic way of organizing the various version of the Grzegorzcyk hierarchy (up to ϵ_0). Notice that the (Ordinal notation) example is one such hierarchy. This system $\lambda\mathbf{G}$ doesn’t have any explicit iteration templates, so these must be simulated in certain ways. The ‘G’ in $\lambda\mathbf{G}$ stands for ‘Gödel’, and also reminds us of Grzegorzcyk.

The system $\lambda\mathbf{H}$ is the term calculus of Howard’s system of constructive ordinals (which is where the ‘H’ comes from). It is an extension of $\lambda\mathbf{G}$ by the addition of explicit iteration templates, and the various gadgetry needed to control these.

In the remainder of this introduction I will survey what happens in each of the remaining sections.

(2) The syntactic machinery of $\lambda\mathbf{H}$ and $\lambda\mathbf{G}$ is described in Section 2. This, when appropriately expanded, answers the (Syntax) question.

(3) Of course, Section 2 may not be the best place to start reading, for some of the development depends on the intended interpretation of the system, which is described in Section 3. This, when appropriately expanded, answers the (Semantics) question.

It is not clear to me which of Sections 2 and 3 should be read before the other. (In $[\lambda\mathbf{H}]$ the semantic aspects are discussed before the syntactic mechanisms.) You should attempt to read these two sections in parallel (which, in practice, means interlacing the reading).

(4) Section 4 describes a crucial trick that is used throughout the development. It has both a semantic and a syntactic form. Is there a canonical way of lifting a function

$$\mathcal{A} : (\mathbb{N} \longrightarrow \mathbb{A}) \longrightarrow \mathbb{A}$$

to a function

$$\mathcal{A}' : (\mathbb{N} \longrightarrow \mathbb{A}') \longrightarrow \mathbb{A}'$$

on the successor function space \mathbb{A}' of \mathbb{A} ? It turns out that there is one such lifting which, for what we do here, seems quite natural.

(5) Section 5 gives more information concerning the semantic aspects of iteration. However, that material is not important for the rest of the survey, and the section could be skipped and left for later.

(6) Suppose you have some familiarity with Sections 2, 3, and 4. It is then time to address the problem of the evaluation process.

Consider a term $\vdash \alpha : \mathcal{O}$ (in $\lambda\mathbf{H}$) and a structure \mathfrak{A} as above. We claim that α is an iteration template and \mathfrak{A} is a place where such an iteration can be carried out. Thus α should determine an element $\mathfrak{A}(\alpha)$ of the carrier \mathbb{A} of \mathfrak{A} . How do we ‘calculate’ this element? The problem is that α may refer to many higher order gadgets, and may not be the most economical description of the intended iteration. We have to find ways of carrying out the instructions embedded in α . Sections 7, 8, 9, and 10 are concerned with this problem. As a preamble to this Section 6 considers a much smaller systems of iteration templates. These are essentially algebraic in nature and much easier to handle. However, a look at them does give us a feel for what can happen later.

(7) The system $\lambda\mathbf{G}$ has no explicit names for iterations (apart from finite iterations). Nevertheless, as is well known, certain iteration can be simulated within the system. The simpler dressed ordinal below the value ϵ_0 can be simulated using numeric gadgets. We review how this is done in Section 7. This material is not necessary for what comes later, but it does give us a starting point. It also shows how, even within $\lambda\mathbf{G}$, we can move from ordinal notations to the more general iteration templates.

(8) I have described $\lambda\mathbf{H}$ as an applied λ -calculus, but this is not strictly correct. In its purest form a λ -calculus provides names for certain entities and a reduction mechanism by which (some) terms can be reduced to a normal form. In most semantic interpretations two terms with the same normal form become equal. However, there may be terms which name the same entities in all acceptable semantics, but which can not be recognized as ‘equal’ by mere reduction. This is the case with $\lambda\mathbf{G}$, but there it doesn’t matter too much (at least for what we do here). It does matter with $\lambda\mathbf{H}$, for part of the evaluation process is to convert an iteration template into a more convenient form. To handle this we need to enrich $\lambda\mathbf{H}$ with a modicum of equational reasoning. This is done in Section 8. In short, we add the ω -rule to the calculus. Once we have this we can introduce the idea

of a ‘canonical’ template, one for which the evaluation process is comparatively straight forward.

(9) How can we convert an arbitrary iteration template into a canonical template? We need to analyse how various compounds of iterations can be replaced by an equivalent single iteration. This is the topic of Section 9. We see how much, but not all, of the standard arithmetic of ordinals lifts quite naturally to iteration templates.

(10) By this stage we are in a position to start naming particular iteration in $\lambda\mathbf{H}$. At the beginning of Section 10 we show how the simulated iterations of $\lambda\mathbf{G}$ can be replaced by explicit terms of $\lambda\mathbf{H}$. Of course, in terms of ordinals, this doesn’t get us above ϵ_0 , but now we can take off. We find that these explicit versions of the $\lambda\mathbf{G}$ -iteration use only a small part of the facilities available in $\lambda\mathbf{H}$. Roughly speaking only low level types are involved. We conclude Section 10 by producing a battery of iteration templates of higher complexity.

Recall that for each set \mathbb{A} we put

$$\mathbb{A}' = (\mathbb{A} \longrightarrow \mathbb{A})$$

to produce the function space of \mathbb{A} . This construction can be repeated $\mathbb{A}, \mathbb{A}', \mathbb{A}'', \mathbb{A}''', \dots$ at least up the finite levels. Thus we set

$$\mathbb{A}^{(0)} = \mathbb{A} \quad \mathbb{A}^{(r+1)} = \mathbb{A}^{(r)'}$$

for each $r < \omega$ to produce a tower of higher level function spaces on \mathbb{A} which form a kind of backbone for the full function space hierarchy on \mathbb{A} . This construction can be mimicked in $\lambda\mathbf{H}$. In particular, we set

$$\mathcal{O}^{(0)} = \mathcal{O} \quad \mathcal{O}^{(r+1)} = \mathcal{O}^{(r)'}$$

for each $r < \omega$ to produce a tower of higher level types on \mathcal{O} . Here for each type τ we have

$$\tau' = (\tau \rightarrow \tau)$$

where ‘ \rightarrow ’ is the syntactic analogue of the function space construction.

In Section 10 we produce a family of terms

$$\vdash [l] : \mathcal{O}^{(l+2)}$$

(for $l < \omega$), and these are combined to produce a family of iteration templates

$$\vdash \Delta[r] : \mathcal{O}$$

(for $r < \omega$). The complexity of these templates increase with r .

I have referred to the complexity of an iteration template before. What can this mean?

To explain that think again of $\lambda\mathbf{G}$ and its connection with the Grzegorzcyk hierarchies. Each term $\vdash \ulcorner g \urcorner : \mathcal{N}'$ of $\lambda\mathbf{G}$ names a function $g : \mathbb{N}'$ (where again the $(\cdot)'$ abbreviations has been used twice, and here we have distinguished between the function g and its name $\ulcorner g \urcorner$). How do we measure the complexity of g ?

One quite useful way is to fix a base function f and generate a Grzegorzcyk hierarchy f_α indexed by an initial stretch of ordinals. We then find the α for which g is ‘most like’

f_α (where ‘most like’ usually means something to do with rate of growth). This gives an ordinal valued measure of complexity.

This measure is relative to the choice of the base function f and the family of notations used to generate the ordinals. There is also a more intrinsic measure of complexity which can be read off from the name $\lceil g \rceil$. We inspect this term to see just which syntactic gadgetry is used in its construction. In particular, we look at the level of types used, and which forms of recursion are needed. There is a nice correspondence between this kind of syntactic measure and the previous ordinal measure. The syntax immediately gives us an upper bound on the ordinal. (For instance, we observe the common phenomenon that a raise in one type level causes an exponential increase in ordinal value.) Furthermore, for each ordinal there is a certain amount of syntax which must be used in the name of at least one function of that ordinal complexity.

(I have been told a story that when Gentzen first tried to explain to Hilbert and his crew the connection between type jump and ordinal exponentiation he was met with more than mild disinterest. I find that hard to believe, but it could be true.)

We can try similar methods to measure the complexity of iteration templates.

Each term $\vdash \alpha : \mathcal{O}$ produces an element $\mathfrak{A}(\alpha)$ of each structure \mathfrak{A} with the appropriate furnishings. The ordinals provide such a structure

$$\mathfrak{Ord} = (\text{Ord}, 0, S, \bigvee)$$

where Ord is the countable ordinal, and the furnishings are obvious. Thus each template α has an ordinal value $\mathfrak{Ord}(\alpha)$, which must give some indication of complexity. Of course, something will be lost in the step $\alpha \mapsto \mathfrak{Ord}(\alpha)$, otherwise we could retrieve α from its value. (That can’t be done even for the value ω . You have no idea which fundamental sequence $\omega[\cdot]$ I am thinking of.)

In particular, each term $\Delta[r]$ (produced in Section 10) has an ordinal value $\mathfrak{Ord}(\Delta[r])$ obtained by performing the iteration $\Delta[r]$ in the ordinals. The rest of the survey, from Section 11 onwards, is concerned with describing these ordinals. As we calculate these values we begin to see another connection between type level and ordinal value. Thus there is a possibility that the very syntax of $\lambda\mathbf{H}$ can be used to measure the complexity of iteration templates and indeed ordinals.

(11) Section 11 can be read independently of the previous development. Much of the standard material on ordinal notations is concerned with extracting the fixed points of normal functions. In this section we show how that kind of material can be lifted to higher levels, making use of higher level functions. The trick is to focus on those high level functions that generate normal functions by iteration, the helpful functions. This section is a condensed version of [FandH], and should be read in conjunction with that paper.

(12) As explained earlier the primary job of $\lambda\mathbf{H}$ is to name \mathbb{O} -based gadgets. However, some of these names can be ‘interpreted’ as Ord -based gadgets. Each term $\vdash \alpha : \mathcal{O}$ immediately gives us $\mathfrak{D}(\alpha) \in \mathbb{O}$. By passing this across a certain canonical function

$$\mathbb{O} \longrightarrow \text{Ord}$$

(called the mediating arrow) we obtain $\mathfrak{Ord}(\alpha) \in \text{Ord}$. However, this method does not extend to higher order terms. Thus, for instance, it does not enable us to interpret a general term $\vdash t : \mathcal{O}''$ (say) as a member of Ord'' . Nevertheless, some closed terms of

λH can be ‘interpreted’ over Ord (as well as over \mathbb{O}). Section 12 shows how this can be done for some of the important terms produced earlier.

The remaining sections are a kind of historical survey of the idea of ordinal notations. They can be viewed as a kind of motivation for the earlier parts of this survey. In fact, it is not a bad idea to look at these section before getting into the earlier stuff.

(13) It could be argued that the topic of ordinal notations started with Veblen’s paper [26]. It is from there that we get the crucial notion of a normal function (although Veblen didn’t use that terminology), and the idea of enumerating the fixed points of such functions to produce faster normal functions. By iterating that idea Veblen produced his first rather short hierarchy, and then by iterating that construction he produced a much longer hierarchy. These days this technique is usually described as a kind of Cantor-Bendixson process on sets of ordinals. We hit a set of ordinals with a certain derivative to produce a sparser set of ordinals, and then we iterate this process. In fact, if we return to Veblen’s original idea we find it is really quite flexible. Section 13 first describes the Veblen hierarchies in more or less his terms. Then the hierarchies are translated into related hierarchies using the helpful functions of Section 11. Once we do that we see it is quite easy to organize Veblen’s ideas and go far beyond the ordinals he obtained. The latter parts of Veblen’s paper are somewhat obscure. The re-working in terms of helpful functions does clear away some of the mist.

(14) In the first instance a Veblen hierarchy is quite short. Veblen’s idea was to generate many such hierarchies which interact in certain quite intricate ways. We need a method of organizing this whole family. The crucial trick is due to Bachmann. We use certain *uncountable* ordinals. Section 14 sets down the required background to do this. Once we get into the details we see that the uncountable ordinal are really just a convenient way of describing patterns, or templates, for constructing certain functions. The ordinal properties are hardly important.

(15) Section 11 gives us a stock of basic helpful functions. These can be combined in several ways to produce a vast collection of helpful functions. Using the ideas of Section 14 the construction of each such compound function can indexed by an uncountable ordinal. Section 15 describes how this is done. This material is related to parts of [Com] and the connections are described.

(16) So far the idea has been to develop ordinal (or iteration) notations from below. Already generated ordinals are used to index a process of generating larger ordinals, which are then used to produce even larger ordinals, and so on. The current preferred development is from above. Building on Bachmann’s idea a function from uncountable ordinals to countable ordinal is produced. Section 16 shows how the two methods are connected. This is related to the main result of [Com]. Here certain results omitted from that paper are given.

(17) The enumerating function used in Section 16 seems simpler than the ones found in the literature. This section shows that, in fact, they are the same.

There are many other works on ordinal notations, and some of these can be seen in the more general context of iteration templates. Various references are given as the material is developed. The papers [1] and [13] are worth a special mention since they use higher order functionals.

The work of Danner in [10] and [11] is particularly relevant. As here applied λ -calculi are used to name various ordinal gadgets. Furthermore, Danner develops a lot of the

associated proof theoretic material, something that is hardly touched on here.

2 The systems $\lambda\mathbf{G}$ and $\lambda\mathbf{H}$

In this section we set up most of the syntactic mechanisms of $\lambda\mathbf{G}$ and $\lambda\mathbf{H}$. Both of these are applied λ -calculi and $\lambda\mathbf{G}$ is a part of $\lambda\mathbf{H}$. We won't spend too much time on the routine stuff, for this is mostly well known and can be found in [25]. Similarly, we won't spend too much time (in this section) describing what the constructions are suppose to mean. Again, the intended meanings for $\lambda\mathbf{G}$ are well known (and given in [25]), and most of the rest of this survey is about how we can extract information from $\lambda\mathbf{H}$.

The system $\lambda\mathbf{G}$ is an applied λ -calculus forming the term calculus of Gödel's T . It is designed to name many functions in the function space hierarchy on \mathbb{N} , but, unlike Gödel's T it has no reasoning facilities (so the question of when two functions are equal can not even be formulated in $\lambda\mathbf{G}$, never mind answered). Descriptions of $\lambda\mathbf{G}$ can be found in [24] and [14], with some useful material in [16]. However, these are not explicitly in terms of a λ -calculus. The long survey paper [3] is about Gödel's T and contains a large amount of information.

The system $\lambda\mathbf{H}$ is an applied λ -calculus enriched with a modicum of equational reasoning. It does for Howard's system of constructive ordinals, what $\lambda\mathbf{G}$ does for Gödel's T . That is, it is designed merely to name gadgets rather than reason about them. Howard's system was introduced in [17] with the aim of naming certain ordinals *and* proving that they are ordinals (that is, they are well founded). The original system takes a rather relaxed view towards syntax. More importantly, it encompasses various reasoning principles, some of which are set theoretic. The survey paper [3] also discusses Howard's system. Indeed, Feferman's OR_1^ω described in section 9.1 of [3] is related to Howard's system and to $\lambda\mathbf{H}$. However, OR_1^ω still has quite a lot of reasoning facilities.

As with any applied λ -calculus the two systems $\lambda\mathbf{G}$ and $\lambda\mathbf{H}$ have several syntactic categories such as Type, Term, Typing-judgement, Typing-derivation, Computation, Equational-judgement, and Equational-derivation. Let's begin to describe these.

2.1 DEFINITION. The system $\lambda\mathbf{G}$ has just one atom (ground type) \mathcal{N} .

The system $\lambda\mathbf{H}$ has two atoms (ground types) \mathcal{N} and \mathcal{O} .

For both systems all other types are generated from the atoms by arrow formation. Thus if σ and ρ are types then $(\sigma \rightarrow \rho)$ is a type. ■

(You may think that both systems should also contain product types. You are right. I will say a few words about this at the end of this section.)

Notice that each type of $\lambda\mathbf{G}$ is also a type of $\lambda\mathbf{H}$, In fact, a $\lambda\mathbf{H}$ -type is a $\lambda\mathbf{G}$ -type precisely when it does not contain the atom \mathcal{O} .

The idea behind $\lambda\mathbf{G}$ is clear. The atom \mathcal{N} is a name for the concrete set \mathbb{N} of natural numbers. With this each $\lambda\mathbf{G}$ -type names a particular member of the concrete function space hierarchy over \mathbb{N} . The other syntactic devices of $\lambda\mathbf{G}$ (to be described shortly) are designed to name and manipulate members of these function spaces.

What about $\lambda\mathbf{H}$? The idea behind the $\lambda\mathbf{G}$ -part remains the same. In particular, \mathcal{N} is the name of \mathbb{N} . However, the intended interpretation of \mathcal{O} is not so clear. We are trying to make precise the notion of a **concrete iteration template**. The atom \mathcal{O} is a name for the set \mathbb{O} of all these, and at the moment this is a rather hazy idea. The approach we take

is to set down several properties we believe \mathbb{O} should have and then attempt to extract from these further information which, we hope, will make our view of \mathbb{O} a bit sharper. The system λH is designed to organize these properties.

As indicated we use lower case Greek letters $\rho, \sigma, \tau, \dots$ for types. We employ the usual conventions for omitting brackets, namely that which matches well with currying. For each type τ we let

$$\tau' \text{ abbreviate } \tau \rightarrow \tau$$

to produce the simplest symbolic function space attached to τ . This construction can be iterated. Thus

$$\tau'' \text{ abbreviate } (\tau \rightarrow \tau) \rightarrow (\tau \rightarrow \tau)$$

and $\tau''', \tau^{iv}, \tau^v, \dots$ are higher order compounds of τ . For each type σ let

$$\mathcal{L}(\sigma) \text{ abbreviate } (\mathcal{N} \rightarrow \sigma) \rightarrow \sigma$$

using the atom \mathcal{N} . Don't worry about why we do this, it will become clear in time.

The terms of a λ -calculus are designed to name inhabitants of types. They are built up in certain ways from a fixed collection of constants.

2.2 DEFINITION. The system λH has five constants with housing axioms

$$\underline{0} : \mathcal{N} \quad \underline{S} : \mathcal{N}' \quad \bar{0} : \mathcal{O} \quad \bar{S} : \mathcal{O}' \quad \text{Lim} : \mathcal{L}(\mathcal{O})$$

and two families of constants

$$I_\sigma : \mathcal{N} \rightarrow \sigma'' \quad J_\sigma : \mathcal{O} \rightarrow \mathcal{L}(\sigma) \rightarrow \sigma''$$

indexed by the types.

The system λG has the constants $\underline{0}, \underline{S}$, and the constants I_σ where σ is a λG -type. ■

The idea behind these takes a bit more explaining. To do that let's introduce some terminology.

2.3 DEFINITION. Assume given an unlimited stock of identifiers. With these we generate the **raw terms** as follows.

Each constant (as given in Definition 2.2) and each identifier is a raw term. If q and p are raw terms then (qp) is a raw term. If r is a raw term, σ a type, and y an identifier, then $(\lambda y : \sigma . r)$ is a raw term.

We often say 'term' when we mean 'raw term'. ■

(Identifiers are often called variables, but this can be confusing when there are variable atomic types around.)

These raw terms are nothing more than the typed λ -terms built up from the given constants. They are 'raw' because as yet we have imposed no typing discipline on their construction. In particular, some of these terms can not possibly have a meaning since they are 'grammatically incorrect'.

Brackets are put into terms to ensure that each is uniquely parsed. Often we informally omit brackets from a term in the usual way to match with currying.

For each pair of terms t, s and each $m \in \mathbb{N}$ we generate a term $t^m s$ by

$$t^0 s = s \quad t^{m+1} s = t(t^m s)$$

and we refer to $t^m s$ as the m -fold iterate of t on s . Using this notation we set

$$\underline{m} = \underline{S}^m \underline{0} \quad \overline{m} = \overline{S}^m \overline{0}$$

to produce two families of compound terms.

What's the idea behind the constants of Definition 2.2. At this stage we can certainly motivate the $\lambda\mathbf{G}$ -constants, but the others are harder to explain.

The constant $\underline{0}$ is the name of the natural number 0, and the constant \underline{S} is the name of the successor function S on \mathbb{N} . Using these the numeral

$$\underline{m} = \underline{S}^m \underline{0}$$

is the canonical name of the natural number m . The constants $\overline{0}, \overline{S}$ and the compounds \overline{m} do a similar job for concrete iteration templates. (At this stage you might not see the intended difference in meaning between \underline{m} and \overline{m} . In the final analysis distinguishing between these is a bit of pedantry, but at this stage it is a useful bit of pedantry for it helps to avoid some confusion later.)

What does the constant l_σ do? Consider any concrete set \mathbb{S} . For each $m \in \mathbb{N}$, each function $f \in \mathbb{S}' = (\mathbb{S} \longrightarrow \mathbb{S})$, and each argument $x \in \mathbb{S}$, we can form

$$f^m x \in \mathbb{S}$$

the m^{th} iterate of f applied to x . This operation consumes three arguments m, f, x (in that order) to return the eventual value $f^m x$. The constant l_σ is the name of such an operation, which is why we call l_σ an iterator.

The constant J_σ does a similar job for concrete iteration templates. Just how it does this, and how the constant Lim becomes involved is what this survey is about, so don't worry about these for a while.

We remarked above that a raw term may be nonsensical because of its non-grammatical structure. We want to isolate those terms which are grammatically correct (so that later we can consider what they might mean). This is done using a derivation system to impose a typing discipline on terms.

We need a bit of a preamble.

A statement is a pair $t : \tau$ where t is a term and τ is a type. We call t the subject and τ the predicate of the statement $t : \tau$. Thus, in Definition 2.2, each housing axiom is a statement. In other words, each constant comes with its own housing type. A declaration is a statement $x : \sigma$ where x is an identifier. A context is a list

$$\Gamma = x_1 : \sigma_1, \dots, x_l : \sigma_l$$

of declarations. Such a context Γ is legal if the declared identifiers x_1, \dots, x_l are distinct. A judgement (or Typing-judgement in full)

$$\Gamma \vdash t : \tau$$

is a statement $t : \tau$ in context Γ . We wish to read this as

Within the *legal* context Γ the *well-formed* term t inhabits the *acceptable* type τ

but such a reading requires some justification. For us every type is acceptable, and the legality of a context is easy to detect. The well-formedness of a term is more complicated. We set up a formal system to generate those judgements we wish to isolate.

2.4 DEFINITION. A derivation (or Typing-derivation in full)

$$(\nabla) \quad \Gamma \vdash t : \tau$$

is a finite rooted tree of judgements grown according to a certain collection of rules. ■

We need not set down these rules here, for we don't need the details. There are just the standard rules for λ -calculi, and are given in Table 1 of [λH], and are worried to death in [25]. However, a small example will at least keep us amused.

2.5 EXAMPLE. Let's have a look at the original 3-variable Ackermann function. This is the function ACK recursively specified by

$$\begin{aligned} ACK(0, 0, m) &= m + 1 \\ ACK(i + 1, 0, m) &= ACK(i, m, m) \\ ACK(i, r + 1, m) &= ACK(i, 0, ACK(i, r, m)) \end{aligned}$$

for $i, r, m \in \mathbb{N}$. How can we name this in $\lambda\mathbf{G}$? Because we don't have explicit product types we name the curried form of ACK , that is we view it as a function

$$ACK : \mathbb{N} \longrightarrow \mathbb{N} \longrightarrow \mathbb{N}'$$

which consumes its arguments one after another rather than together.

For convenience let

$$I = I_{\mathcal{N}} \quad J = I_{\mathcal{N}'}$$

(to stop the subscripts cluttering up the place). With these let

$$\text{ack} = \lambda y : \mathcal{N}', x : \mathcal{N}. I(\underline{S}x)yx \quad \text{BDY} = I(\underline{S}y)(Jz\text{ack}\underline{S}) \quad \text{ACK} = \lambda z, y, x : \mathcal{N}. \text{BDY}$$

where x, y, z are distinct identifiers. This gives three $\lambda\mathbf{G}$ -terms where the body term BDY contains free identifiers y, z , but the other two terms contain only bound identifiers. It is not immediately obvious that these are well formed. To check this let

$$\Sigma = y : \mathcal{N}', x : \mathcal{N} \quad \Gamma = z : \mathcal{N}, y : \mathcal{N}, x : \mathcal{N}$$

to obtain two legal contexts. With these the derivations of Table 1 show the correctness of the three terms.

After a bit more effort we can show that ACK does name ACK . (We will return to this example later.) ■

When we set up the system $\lambda\mathbf{G}$ and $\lambda\mathbf{H}$ we have in mind that each derivation

$$(\nabla) \quad \Gamma \vdash t : \tau$$

should name a certain concrete gadget. Precisely how this is done in general is described in the next section. Here we can use Example 2.5 to illustrate some of the problems that might arise.

$$\begin{array}{c}
\frac{\frac{\frac{\Gamma \vdash I : \mathcal{N} \rightarrow \mathcal{N}''}{\Gamma \vdash I(\underline{S}x) : \mathcal{N}''} \quad \frac{\frac{\Sigma \vdash \underline{S} : \mathcal{N}' \quad \Sigma \vdash x : \mathcal{N}}{\Sigma \vdash \underline{S}x : \mathcal{N}}}{\Sigma \vdash I(\underline{S}x)y : \mathcal{N}'}}{\Sigma \vdash I(\underline{S}x)yx : \mathcal{N}'} \quad \Sigma \vdash y : \mathcal{N}' \quad \Sigma \vdash x : \mathcal{N}}{\vdash \text{ack} : \mathcal{N}''} \\
\vdots \\
\frac{\frac{\frac{\Gamma \vdash J : \mathcal{N} \rightarrow \mathcal{N}'' \quad \Gamma \vdash z : \mathcal{N}}{\Gamma \vdash Jz : \mathcal{N}'''} \quad \frac{\vdash \text{ack} : \mathcal{N}''}{\Gamma \vdash \text{ack} : \mathcal{N}''}}{\Gamma \vdash Jz\text{ack} : \mathcal{N}''} \quad \Gamma \vdash \underline{S} : \mathcal{N}'}{\Gamma \vdash Jz\text{ack}\underline{S} : \mathcal{N}'} \\
\vdots \\
\frac{\frac{\frac{\frac{\Gamma \vdash I : \mathcal{N} \rightarrow \mathcal{N}''}{\Gamma \vdash I(\underline{S}y) : \mathcal{N}''} \quad \frac{\frac{\Gamma \vdash \underline{S} : \mathcal{N}' \quad \Gamma \vdash y : \mathcal{N}}{\Gamma \vdash \underline{S}y : \mathcal{N}}}{\Gamma \vdash I(\underline{S}y) : \mathcal{N}''} \quad \frac{\vdots}{\Gamma \vdash Jz\text{ack}\underline{S} : \mathcal{N}'}}{\Gamma \vdash \text{BDY} : \mathcal{N}'} \\
\vdots \\
\vdash \text{ACK} : \mathcal{N} \rightarrow \mathcal{N} \rightarrow \mathcal{N}'
\end{array}$$

Table 1: Three example derivations

Consider the term

$$\vdash \text{ACK} : \mathcal{N} \rightarrow \mathcal{N} \rightarrow \mathcal{N}'$$

which, we claim, names a version of the Ackermann function. How can we justify this claim? Given $i, r, m \in \mathbb{N}$ we can use the numerals $\underline{i}, \underline{r}, \underline{m}$ to form

$$\vdash \text{ACK} \underline{i} \underline{r} \underline{m} : \mathcal{N}$$

which is a term that ought to name a natural number, the value $n = \text{ACK}(i, r, m)$. However, the term $\text{ACK} \underline{i} \underline{r} \underline{m}$ is nothing like the numeral \underline{n} (from which we could immediately read off the value). Thus there must be some relationship between the two terms

$$\text{ACK} \underline{i} \underline{r} \underline{m} \quad \underline{n}$$

to be uncovered. The connection, of course, is that \underline{n} is the normal form of $\text{ACK} \underline{i} \underline{r} \underline{m}$ and we can move from the complex term to the numeral by a process of reduction.

Each applied λ -calculus has an associated reduction mechanism base on β -reduction augmented in a certain way to handle the constants. This gives a reduction relation

$$t^- \triangleright t^+$$

between terms generated from certain primitive, 1-step reductions

$$t^- \triangleright t^+$$

postulated at the outset. These 1-step reductions always include redex removal

$$(\lambda x : \sigma . r)s \triangleright r[x := s]$$

which involves substituting a term s for all the free occurrence of some identifier x in some other term r . We need not go into the details here. The other 1-step reductions are often related to the constants, as in our case.

2.6 DEFINITION. For each type σ the constants I_σ and J_σ have the 1-step reductions

$$\begin{array}{ll} I_\sigma 0ts & \triangleright s & J_\sigma \bar{0}lts & \triangleright s \\ I_\sigma (\underline{S}u)ts & \triangleright t(I_\sigma uts) & J_\sigma (\bar{S}\alpha)lts & \triangleright t(J_\sigma \alpha lts) \\ & & J_\sigma (\text{Lim } p)lyx & \triangleright l(\lambda u : \mathcal{N} . J_\sigma (pu)lts) \end{array}$$

where s, t, l, u, α, p are arbitrary terms. ■

As we will see, these capture the intended meaning of I_σ and J_σ as iteration gadgets over \mathcal{N} and \mathcal{O} , respectively. We can't yet explain the use of J_σ , but we can get an inkling into the use of I_σ .

2.7 EXAMPLE. Consider first an arbitrary iterator I_σ and two terms s, t . Consider also $m \in \mathbb{N}$ and its numeral \underline{m} . By repeated use of the 1-step reductions for I_σ we obtain

$$I_\sigma \underline{m}ts \triangleright t^m s$$

so that we may say that $I_\sigma \underline{m}ts$ somehow encodes the m^{th} iterate $t^m s$ of t on s .

Now let's go back to Example 2.5. Now consider $i, r, m \in \mathbb{N}$ with associated numerals $\underline{i}, \underline{r}, \underline{m}$. We have

$$J_{\underline{i}} \text{ack } \underline{S} \triangleright \text{ack}^{\underline{i}} \underline{S}$$

and then

$$\text{ACK } \underline{i} \underline{r} \underline{m} \triangleright \underline{r+1} (J_{\underline{i}} \text{ack } \underline{S}) \underline{m} \triangleright (\text{ack}^{\underline{i}} \underline{S})^{\underline{r+1}} \underline{m}$$

which begins to explain why ACK names ACK. There is a functional $ack : \mathbb{N}''$ such that

$$\text{ACK}(i, r, m) = (\text{ack}^i S)^{r+1} m$$

hold for all $i, r, m \in \mathbb{N}$. ■

To conclude this section let's clear up a minor point that may have been puzzling you.

Usually Gödel's T is presented in terms of recursion; certain recursors are used where we have iterators. Two questions come to mind (which I have rarely seen addressed in print). What is so special about the form of the recursors used? Doesn't the use of mere iterators weaken the system? (The answers are : Nothing, and Not really.)

The types, as given by Definition 2.1, are generated from the atoms using only arrow formation. In a more detailed development we would also include product formation, if σ and ρ are types then $(\sigma \times \rho)$ is a type. This construct also has various associated constants and 1-step reductions. For each pair σ, ρ we must put in the two projections and the pair former.

With product types we can perform two iterations in parallel, where each may interfere with the other. Using this we can simulate all the recursions we ever need.

It is certainly true that, in contrast with more general recursion, iteration is easier to handle. Also, in this survey, we do not need anything beyond iteration. Thus, to simplify the account, I have left out any discussion of product types.

3 The semantics of $\lambda\mathbf{G}$ and $\lambda\mathbf{H}$

How can we give a meaning to the syntactic constructions of $\lambda\mathbf{G}$ and $\lambda\mathbf{H}$? The intention is that each closed term should name a concrete gadget. Usually this is a function of some kind, but a term

$$\vdash t : \mathcal{N} \quad \vdash t : \mathcal{O}$$

should name, respectively, a natural number or a 'concrete iteration template' whatever that is. Of course, to determine these named gadgets we have to attach a meaning to a wider class of syntactic constructs, the derivable judgement and, in the first instance, the derivations themselves.

The intended semantics for $\lambda\mathbf{G}$ is clear. We work in \mathbf{Set} , the category of sets and functions. The atom \mathcal{N} names \mathbb{N} , the set of natural numbers, and the rest follows in a routine fashion. Normally for this simple case we do not need to spell out how this is done. However, here we will look at some of the details for at least two reasons.

A semantics for $\lambda\mathbf{G}$ can be given in any cartesian closed category \mathbf{Mod} (thought of as the modelling category) with a selected number object \mathbb{N} (which need not be the actual natural numbers). The iterators are obtained from certain mediating properties of \mathbb{N} .

Even if we work entirely within \mathbf{Set} (as we do here) it is not clear what the interpretation of the atom \mathcal{O} should be. The intended meaning of \mathcal{O} is *not* the set $\mathbb{O}rd$ of (countable) ordinals, or some initial stretch of these. It is something wider than that, but what? The more general properties of \mathbb{N} show us how we can find an interpretation \mathbb{O} of \mathcal{O} , and leads us to the interpretation of the iterators.

Let's quickly review the construction of a categorical semantics with an eye on how we select \mathbb{O} .

We choose a cartesian closed category \mathbf{Mod} which in our case will be \mathbf{Set} . We select two objects \mathbb{N} and \mathbb{O} in a manner to be described shortly. Using these we attach a object $\llbracket \tau \rrbracket$ to each type τ . Thus we set

$$\llbracket \mathcal{N} \rrbracket = \mathbb{N} \quad \llbracket \mathcal{O} \rrbracket = \mathbb{O} \quad \llbracket \sigma \rightarrow \rho \rrbracket = \llbracket \sigma \rrbracket \multimap \llbracket \rho \rrbracket$$

using the internal arrow objects of \mathbf{Mod} . This gives a meaning to each type. (Notice that explicit product types can be handle in the same way.) Continuing, each context

$$\Gamma = x_1 : \sigma_1, \dots, x_l : \sigma_l$$

receives a meaning

$$\llbracket \Gamma \rrbracket = (\dots (\mathbf{1} \times \llbracket \sigma_1 \rrbracket) \times \dots) \times \llbracket \sigma_l \rrbracket$$

using the product objects of \mathbf{Mod} . Here $\mathbf{1}$ is the final object of \mathbf{Mod} , and is the interpretation of the empty context.

With these we are almost in a position to attach to each derivation

$$(\nabla) \quad \Gamma \vdash t : \tau$$

an arrow

$$\llbracket \Gamma \rrbracket \xrightarrow{\llbracket \nabla \rrbracket} \llbracket \tau \rrbracket$$

of \mathbf{Mod} . This is the meaning of the derivation, the gadget named by the derivation. We attach the arrow by tracking through ∇ from leaves to root to match internal constructs of \mathbf{Mod} with the syntactic constructs in ∇ . The only problem is to interpret those leaves which use a constant. We will look at that shortly.

Each derivation ∇ produces an arrow $\llbracket \nabla \rrbracket$ in \mathbf{Set} , our chosen modelling category. The source of this arrow is determined by the context of the root judgement of ∇ . In particular, each derivation

$$(\nabla) \quad \vdash t : \tau$$

in the empty context produces an arrow

$$\mathbf{1} \xrightarrow{\llbracket \nabla \rrbracket} \llbracket \tau \rrbracket$$

where $\mathbf{1}$ is the 1-element set. This is just a fancy way of selecting an element of $\llbracket \tau \rrbracket$. It can be shown (but not here) that this element depends only on the normal form of the root term t and not the particular derivation used to witness the syntactic correctness of t . We write $\llbracket t \rrbracket$ for this element.

As a particular case of this each closed term

$$\vdash \alpha : \mathcal{O}$$

determines some element $\llbracket \alpha \rrbracket \in \mathcal{O}$. We think of α as a syntactic description of the concrete iteration gadget $\llbracket \alpha \rrbracket$. Different α can give the same $\llbracket \alpha \rrbracket$, so there is the problem of which syntactic iteration templates have the same meaning. Later in this survey we develop some arithmetic to help with this.

Let's now return to the problem of determining the interpretation of each iterator. We deal with the \mathcal{N} -iterators l_σ first.

3.1 DEFINITION. A **peano structure** $\mathfrak{A} = (\mathbb{A}, a, A)$ is a set \mathbb{A} furnished with a distinguished element $a \in \mathbb{A}$ and a 1-placed operation $A : \mathbb{A}'$.

A peano arrow

$$\mathfrak{B} \xrightarrow{\phi} \mathfrak{A}$$

from a peano structure $\mathfrak{B} = (\mathbb{B}, b, B)$ to a peano structure \mathfrak{A} is a function $\phi : \mathbb{B} \rightarrow \mathbb{A}$ where $\phi b = a$ and $\phi \circ B = A \circ \phi$.

These objects and arrows form the category **Peano**. ■

Working over **Set** the category has an initial object

$$\mathfrak{N} = (\mathbb{N}, 0, S)$$

which is the natural numbers \mathbb{N} furnished with zero, 0 , and the successor operation, S . For each peano structure $\mathfrak{A} = (\mathbb{A}, a, A)$ there is a unique peano arrow

$$\mathfrak{N} \longrightarrow \mathfrak{A}$$

that is a unique function

$$I_{\mathfrak{A}} : \mathbb{N} \longrightarrow \mathbb{A}$$

such that

$$I_{\mathfrak{A}}0 = a \quad I_{\mathfrak{A}}(Su) = A(I_{\mathfrak{A}}u)$$

for each $u \in \mathbb{N}$. Each set \mathbb{A} may carry many different peano structures $\mathfrak{A} = (\mathbb{A}, a, A)$. By uncoupling the furnishings we obtain a function

$$I_{\mathbb{A}} : \mathbb{N} \longrightarrow \mathbb{A}''$$

where

$$I_{\mathbb{A}}uAa = I_{\mathfrak{A}}u$$

for each such \mathfrak{A} and $u \in \mathbb{N}$. This function $I_{\mathbb{A}}$ satisfies

$$I_{\mathbb{A}}0Aa = a \quad I_{\mathbb{A}}(Su)Aa = I_{\mathfrak{A}}(Su) = A(I_{\mathfrak{A}}u) = A(I_{\mathbb{A}}uAa)$$

for each $u \in \mathbb{N}$, and so $I_{\mathbb{A}}$ provides the interpretation for the iterator l_{σ} when $[\sigma] = \mathbb{A}$.

All this follows from the initiality of \mathfrak{N} in the category **Peano**. We use the same ideas to produce \mathbb{O} and its associated facilities. However, this time the results are not obvious.

We need an analogue of **Peano**.

3.2 DEFINITION. A limit creator over a set \mathbb{A} is a function $\mathcal{A} : \mathcal{L}(\mathbb{A})$ where

$$\mathcal{L}(\mathbb{A}) = (\mathbb{N} \longrightarrow \mathbb{A}) \longrightarrow \mathbb{A}$$

is a higher order function space. ■

As indicated, each limit creator $\mathcal{A} : \mathcal{L}(\mathbb{A})$ converts each sequence $p : \mathbb{N} \longrightarrow \mathbb{A}$ through \mathbb{A} into some element $\mathcal{A}p \in \mathbb{A}$. For many natural situations this *is* some kind of limit of the sequence, but we also allow some rather strange examples. To help us remember this we sometimes say \mathcal{A} **collates** the sequence p , or $\mathcal{A}p$ is the **collation** of p (rather than the limit).

3.3 DEFINITION. A limit structure $\mathfrak{A} = (\mathbb{A}, a, A, \mathcal{A})$ is a set \mathbb{A} furnished with a distinguished element $a \in \mathbb{A}$, a 1-placed operation $A : \mathbb{A}'$, and a limit creator $\mathcal{A} : \mathcal{L}(\mathbb{A})$.

A limit arrow

$$\mathfrak{B} \xrightarrow{\phi} \mathfrak{A}$$

from a limit structure $\mathfrak{B} = (\mathbb{B}, b, B, \mathcal{B})$ to a limit structure \mathfrak{A} is a function $\phi : \mathbb{B} \longrightarrow \mathbb{A}$ where $\phi b = a$, $\phi \circ B = A \circ \phi$, and $\phi \circ \mathcal{B} = \mathcal{A} \bullet \phi$. Here

$$(\mathcal{A} \bullet \phi)p = \mathcal{A}(\phi \circ p)$$

for each $p : \mathbb{N} \longrightarrow \mathbb{B}$.¹

These objects and arrows form the category **Limit**. ■

¹This condition merely says that ϕ commutes with ‘taking limits’.

The proof that **Limit** is a category is not immediate, but soon follows by a small calculation. We will look at the details of this and other aspects of **Limit** in Section 5, but now it's time for some examples.

3.4 EXAMPLES. (a) For a set \mathbb{A} let $\mathcal{P}_\omega \mathbb{A}$ be the set of countable subsets of \mathbb{A} . Suppose we have some kind of collection operation

$$\bigvee : \mathcal{P}_\omega \mathbb{A} \longrightarrow \mathbb{A}$$

on \mathbb{A} . We can think of as a supremum operation, but there are more exotic examples. In particular, we don't need a comparison relation on \mathbb{A} . Setting

$$\mathcal{A}p = \bigvee \{pu \mid u \in \mathbb{N}\}$$

for each $p : \mathbb{N} \longrightarrow \mathbb{A}$ gives a limit creator $\mathcal{A} : \mathcal{L}(\mathbb{A})$. In practice we often write \bigvee for this particular \mathcal{A} .

As an example of this let $\mathbb{O}rd$ be the set of countable ordinals, and consider

$$\mathfrak{O}rd = (\mathbb{O}rd, 0, S, \bigvee)$$

where 0 is the zero ordinal, S is the ordinal successor function, and \bigvee is the supremum function (for countable sets of countable ordinals). This gives a limit structure which has been investigated for over 100 years. We will use this object as an illustration several times but, of course, we are interested in more than just ordinals.

(b) For a second example let's start with a concrete case. Consider the set $\mathbb{A} = \mathbb{N}'$ of functions on \mathbb{N} . Many ordinal-indexed families of such functions have the form

$$f_0 = f \quad f_{\alpha+1} = Ff_\alpha \quad f_\lambda u = f_{\lambda[u]}u$$

where $f : \mathbb{N}'$ is a base function, $F : \mathbb{N}''$ is a jump functional, α is a countable ordinal, λ is a countable limit ordinal with a selected fundamental sequence $\lambda[\cdot]$, and $u \in \mathbb{N}$. Of course, the phrase 'ordinal-indexed' is wrong since the result can depend quite dramatically on which fundamental sequences are selected. A typical example of a jump functional is the Ackermann jump **ack** given by

$$\mathbf{ack}gx = g^{u+1}u$$

for $g : \mathbb{N}'$ and $u \in \mathbb{N}$. Here g^m indicates the m^{th} iterate of f . This is the jump functional which generates Example 1.1(Ordinal notation). There are also several other quite commonly used jump functionals.

Let $\Delta : \mathcal{L}(\mathbb{N}')$ be the limit creator given by

$$\Delta pu = puu$$

for each $p : \mathbb{N} \longrightarrow \mathbb{N}'$ and $u \in \mathbb{N}$. (This is often called the diagonal limit creator.) With this we see that

$$\mathfrak{G} = (\mathbb{N}', f, F, \Delta)$$

is a limit structure. The ordinal-indexed family of functions traces out a path through this structure as determined by the selected fundamental sequences. If we use different selections of fundamental sequences then the linear hierarchy starts to spread out.

Aspects of this object has been studied for many years, and we will use it in several illustrations. ■

The initial object of **Peano** is the natural numbers \mathfrak{N} with the obvious furnishings. This is essentially the Dedekind characterization in modern dress.

With a bit of work we can show that **Limit** has an initial object, and it is this that we want to understand.

3.5 DEFINITION. Let

$$\mathfrak{D} = (\mathbb{O}, 0, S, Lim)$$

be the initial object of **Limit**. ■

The details of the existence of \mathfrak{D} are given in Section 5. The proof doesn't tell us much about what \mathfrak{D} looks like. However, it must be some set \mathbb{O} with the indicated furnishings. The members of \mathbb{O} are sometimes called **tree ordinals**, and some aspects of these are discussed in [12].

Do not confuse the furnishings 0 and S of \mathfrak{D} with the corresponding furnishings of \mathfrak{N} ; they are not the same. However, it turns out that \mathbb{N} is a part of \mathbb{O} and on this part the two sets of furnishings agree. In fact, \mathfrak{D} is \mathfrak{N} with an explosion of new elements sitting on top.

The system λH is designed to get inside \mathfrak{D} . We use \mathbb{O} to interpret \mathcal{O} , and the furnishings of \mathfrak{D} to interpret $\bar{0}, \bar{S}, Lim$. But how do we interpret the iterators J_σ ?

Consider any limit structure $\mathfrak{A} = (\mathbb{A}, a, A, \mathcal{A})$. By the initiality there is a unique limit arrow

$$\mathfrak{D} \longrightarrow \mathfrak{A}$$

to \mathfrak{A} . Thus there is a unique function

$$J_{\mathfrak{A}} : \mathbb{O} \longrightarrow \mathbb{A}$$

depending on \mathfrak{A} such that

$$J_{\mathfrak{A}}0 = a \quad J_{\mathfrak{A}}(S\alpha) = A(J_{\mathfrak{A}}\alpha) \quad J_{\mathfrak{A}}(Lim p) = \mathcal{A}(J_{\mathfrak{A}} \circ p)$$

for each $\alpha \in \mathbb{O}$ and $p : \mathbb{N} \longrightarrow \mathbb{O}$. Each set \mathbb{A} may carry many different limit structures $\mathfrak{A} = (\mathbb{A}, a, A, \mathcal{A})$. By uncoupling the furnishings we obtain a function

$$J_{\mathbb{A}} : \mathbb{O} \longrightarrow \mathcal{L}(\mathbb{A}) \longrightarrow \mathbb{A}''$$

depending only on \mathbb{A} where

$$J_{\mathbb{A}}\alpha \mathcal{A}Aa = J_{\mathfrak{A}}\alpha$$

for each such \mathfrak{A} and $\alpha \in \mathbb{O}$. This function $J_{\mathbb{A}}$ satisfies

$$J_{\mathbb{A}}0 \mathcal{A}Aa = a \quad J_{\mathbb{A}}(S\alpha) \mathcal{A}Aa = J_{\mathfrak{A}}(S\alpha) = A(J_{\mathfrak{A}}\alpha) = A(J_{\mathbb{A}}\alpha \mathcal{A}Aa)$$

for each $\alpha \in \mathbb{O}$. For each $p : \mathbb{N} \longrightarrow \mathbb{A}$ it satisfies

$$J_{\mathbb{A}}(Lim p) \mathcal{A}Aa = J_{\mathfrak{A}}(Lim p) = \mathcal{A}(J_{\mathfrak{A}} \circ p) = \mathcal{A}(u \mapsto J_{\mathbb{A}}(pu) \mathcal{A}Aa)$$

since

$$(J_{\mathfrak{A}} \circ p)u = J_{\mathbb{A}}(pu) \mathcal{A}Aa$$

for each $u \in \mathbb{N}$. Thus $J_{\mathbb{A}}$ provides the interpretation for the iterator J_σ when $[[\sigma]] = \mathbb{A}$.

3.6 EXAMPLE. Although the function $J_{\mathbb{A}}$ consumes four arguments (a member of \mathbb{O} and the furnishings of the tacit \mathfrak{A}) we often restrict some of these. In particular, it is often useful to fix the particular limit creator $\mathcal{A} \in \mathcal{L}(\mathbb{A})$, and let the other three arguments vary to some degree. For instance, consider the limit structure

$$\mathfrak{G} = (\mathbb{N}', f, F, \Delta)$$

of Example 3.4(b). Let us fix the diagonal limit creator Δ but let f and F vary freely. Let us also consider only those members of \mathbb{O} which arise from ordinals with a family of selected fundamental sequences. Thus let

$$\mathbb{G}\alpha Ff = J_{\mathbb{G}}\alpha\Delta Ff$$

for these α . This gives

$$\mathbb{G}0Ff = f \quad \mathbb{G}(\alpha + 1)Ff = F(\mathbb{G}\alpha Ff) \quad \mathbb{G}\lambda Ffu = \mathbb{G}\lambda[u]Ffu$$

for each ordinal α , limit ordinal λ with selected fundamental sequence $\lambda[\cdot]$, and $u \in \mathbb{N}$. This is just another description of the ordinal-indexed family of Example 3.4(b), and is sometimes called the long Grzegorzczuk hierarchy. ■

We will return to these and similar examples later.

To conclude this section we introduce what is sometimes a more convenient notation for the mediating function $J_{\mathfrak{A}}$ associated with a limit structure. We write $\mathfrak{A}(\cdot)$ for this mediating arrow. Thus

$$\mathfrak{A}(\cdot) : \mathbb{O} \longrightarrow \mathbb{A}$$

is the unique function satisfying

$$\mathfrak{A}(0) = a \quad \mathfrak{A}(S\alpha) = A(\mathfrak{A}(\alpha)) \quad \mathfrak{A}(\text{Lim } p) = \mathcal{A}(u \mapsto \mathfrak{A}(pu))$$

for each $\alpha \in \mathbb{O}$ and $p : \mathbb{N} \longrightarrow \mathbb{A}$. Here ‘ $u \mapsto \mathfrak{A}(pu)$ ’ is a convenient description of the ω -chain in \mathfrak{A} being collated by \mathcal{A} , and ‘ u ’ ranges over \mathbb{N} .

4 Limit lifting

The slightly novel feature of a limit structure \mathfrak{A} is the limit creator $\mathcal{A} : \mathcal{L}(\mathbb{A})$ whose job is to collate a supplied chain $p : \mathbb{N} \longrightarrow \mathbb{A}$ into a single element $\mathcal{A}p$. In this section we look at a method of turning one such limit creator into another. The way this is done may, at first sight, look a little strange but in the end we will see why it works.

For each set \mathbb{A} we write \mathbb{A}' for the function space $\mathbb{A} \rightarrow \mathbb{A}$. We do this to condense some of the notation which otherwise may become a little cumbersome. Suppose we have a limit creator $\mathcal{A} : \mathcal{L}(\mathbb{A})$ on \mathbb{A} . Is there any ‘natural’ way of converting \mathcal{A} into a limit creator on \mathbb{A}' ?

4.1 DEFINITION. Each limit creator $\mathcal{A} : \mathcal{L}(\mathbb{A})$ has a lift $\mathcal{A}' : \mathcal{L}(\mathbb{A}')$ given by $\mathcal{A}'px = \mathcal{A}q$ where $qu = pux$ for all $p : \mathbb{N} \longrightarrow \mathbb{A}'$, $x \in \mathbb{A}$, and $u \in \mathbb{N}$.

This sets up the lifting operation

$$\uparrow_{\mathbb{A}} : \mathcal{L}(\mathbb{A}) \longrightarrow \mathcal{L}(\mathbb{A}')$$

on \mathbb{A} . ■

Informally we often write

$$\mathcal{A}'px = \mathcal{A}(u \mapsto pux)$$

using ‘ $u \mapsto pux$ ’ to indicate the function q of the definition.

This certainly produces a function $\uparrow_{\mathbb{A}}$ of the indicated type, by why is this ‘natural’? In the final analysis the answer to this is ‘Because it works’, which at this stage is not very satisfactory. Perhaps a couple of examples will help.

4.2 EXAMPLE. In Example 3.4(a) we saw that each operation

$$\bigvee : \mathcal{P}_{\omega}\mathbb{A} \longrightarrow \mathbb{A}$$

gives a limit creator $\mathcal{A} : \mathcal{L}(\mathbb{A})$ where

$$\mathcal{A}p = \bigvee \{pu \mid u \in \mathbb{N}\}$$

for each $p : \mathbb{N} \longrightarrow \mathbb{A}$. For such an \mathcal{A} the order of the elements it collates is irrelevant. Sometimes in particular cases \bigvee is a supremum operation for a comparison on \mathbb{A} , and sometimes it is not.

There is a standard way of lifting such a pseudo-supremum to a pseudo-supremum

$$\dot{\bigvee} : \mathcal{P}_{\omega}\mathbb{A}' \longrightarrow \mathbb{A}'$$

on the function space \mathbb{A}' . We use the ‘pointwise supremum’ given by

$$(\dot{\bigvee} \mathcal{G})x = \bigvee \{gx \mid g \in \mathcal{G}\}$$

for each $\mathcal{G} \subseteq \mathbb{A}'$. In practice this is most often used when \bigvee is a supremum operation for a comparison on \mathbb{A} , and then $\dot{\bigvee}$ is the supremum operation for the pointwise comparison on \mathbb{A}' . We will use such a construction in Section 11.

Even when \bigvee is not a supremum operation, we can still form $\dot{\bigvee}$, and so obtain a limit creator $\dot{\mathcal{A}}$ on \mathbb{A}' . But, for each $p : \mathbb{N} \longrightarrow \mathbb{A}'$ and $x \in \mathbb{A}$ we have

$$\dot{\mathcal{A}}px = \dot{\bigvee} \{pu \mid u \in \mathbb{N}\}x = \bigvee \{pux \mid u \in \mathbb{N}\} = \mathcal{A}(u \longmapsto pux) = \mathcal{A}'px$$

to show that $\dot{\mathcal{A}} = \mathcal{A}'$. ■

This shows that for order insensitive collation the lift is just pointwise supremum, which almost always is what we want.

In the next example we work in a quite general setting.

4.3 EXAMPLE. For each limit structure $\mathfrak{A} = (\mathbb{A}, a, A, \mathcal{A})$ the associated function $\mathfrak{A}(\cdot)$ tracks through \mathbb{A} starting from the given element a . However, it might be more convenient if we could start from an arbitrary element. For instance, in this way we could follow one path by another by making the finish element of the first the start element of the second.

To do this consider the lifted limit structure $\mathfrak{A}' = (\mathbb{A}', id_{\mathbb{A}}, A \circ -, \mathcal{A}')$ on \mathbb{A}' where the new successor operation is just composition with A , and the new limit creator is the lift of the old one. Now consider the two functions

$$\mathfrak{A}(\cdot) : \mathbb{O} \longrightarrow \mathbb{A} \quad \mathfrak{A}'(\cdot) : \mathbb{O} \longrightarrow \mathbb{A}'$$

which track through \mathbb{A} and \mathbb{A}' . We may check that

$$\mathfrak{A}'(\alpha)a = \mathfrak{A}(\alpha)$$

holds for each $\alpha \in \mathbb{O}$. To do that we proceed by induction (on canonical names) and show

$$\mathfrak{A}'(0)a = \mathfrak{A}(0) \quad \mathfrak{A}'(S\alpha)a = \mathfrak{A}(S\alpha) \quad \mathfrak{A}'(\text{Lim } p)a = \mathfrak{A}(\text{Lim } p)$$

for each $\alpha \in \mathbb{O}$ and $p : \mathbb{N} \longrightarrow \mathbb{O}$. Let's look at the details of the third equality. We have

$$\mathfrak{A}'(\text{Lim } p)a = \mathcal{A}'(u \mapsto \mathfrak{A}'(pu))a = \mathcal{A}(u \mapsto \mathfrak{A}'(pu)a) = \mathcal{A}(u \mapsto \mathfrak{A}(pu)) = \mathfrak{A}(\text{Lim } p)$$

which is the required result. Here the first equality uses the *Lim*-clause of the definition of $\mathfrak{A}'(\cdot)$, the second uses the definition of \mathcal{A}' , the third uses the induction hypothesis, and the fourth is *Lim*-clause of the definition of $\mathfrak{A}(\cdot)$. ■

There is an argument for claiming that \mathfrak{A}' is a more important structure than \mathfrak{A} . We won't make that here, for the reasoning will become clear as the story unfolds. Perhaps if you think of a different way of describing

$$\mathfrak{A}'(\beta) \circ \mathfrak{A}'(\alpha)$$

(for $\alpha, \beta \in \mathbb{O}$) then you will spot something.

Lifting can be iterated. From $\mathcal{A} : \mathcal{L}(\mathbb{A})$ we generate

$$\mathcal{A}' : \mathcal{L}(\mathbb{A}'), \mathcal{A}'' : \mathcal{L}(\mathbb{A}''), \mathcal{A}''' : \mathcal{L}(\mathbb{A}'''), \dots$$

where

$$\begin{aligned} \mathcal{A}'px &= \mathcal{A}(u \mapsto pux) \\ \mathcal{A}''qyx &= \mathcal{A}'(u \mapsto quyx) = \mathcal{A}(u \mapsto quyx) \\ \mathcal{A}'''rzyx &= \mathcal{A}''(u \mapsto ruzyx) = \mathcal{A}'(u \mapsto ruzyx) = \mathcal{A}(u \mapsto ruzyx) \\ &\vdots \end{aligned}$$

for all $x : \mathbb{A}, y : \mathbb{A}', z : \mathbb{A}, \dots$ and sequence $p : \mathbb{N} \longrightarrow \mathbb{A}', q : \mathbb{N} \longrightarrow \mathbb{A}'', r : \mathbb{N} \longrightarrow \mathbb{A}''', \dots$ of functions. Note, however, that this $(\cdot)'$ notation omits a lot of information. Thus in full we should write

$$\mathcal{A}' = \uparrow_{\mathbb{A}} \mathcal{A} \quad \mathcal{A}'' = \uparrow_{\mathbb{A}'}(\uparrow_{\mathbb{A}} \mathcal{A}) \quad \mathcal{A}''' = \uparrow_{\mathbb{A}''}(\uparrow_{\mathbb{A}'}(\uparrow_{\mathbb{A}} \mathcal{A})) \quad \dots$$

but the condensed notation rarely leads to confusion.

As well as these concrete lifting gadgets we will also need formal analogues in the languages of $\lambda\mathbf{G}$ and $\lambda\mathbf{H}$.

4.4 DEFINITION. For each type σ let

$$\uparrow_{\sigma} = \lambda l : \mathcal{L}(\sigma), q : \mathcal{N} \rightarrow \sigma', x : \sigma . l(\lambda u : \mathcal{N} . qux)$$

to produce a term $\vdash \uparrow_{\sigma} : \mathcal{L}(\sigma) \rightarrow \mathcal{L}(\sigma')$. ■

This dual use of ' \uparrow ' will not cause confusion. If σ names \mathbb{A} , then the syntactic \uparrow_{σ} names the concrete $\uparrow_{\mathbb{A}}$.

Finally, note that the uniformity of this construction is at the level of *types* not inhabitants. In the end this produces a global restriction on what can be done in $\lambda\mathbf{H}$.

In the next section we look at some of the details of the category **Limit**. On the whole these details are not required for the rest of the survey. Thus, if you are prepared to accept that **Limit** does have an initial object \mathfrak{D} , then you can go straight to Section 6.

5 The category of limit structures

In Section 3 we considered three categories **Set**, **Peano**, **Limit** where the first two are well known but the third is not. It would be nice to know more about **Limit**, and its initial object \mathfrak{D} in particular. In this section I will set down what I do know.²

I suppose we should begin at the beginning.

From Definition 3.3 of a limit arrow it is not immediately obvious that these are closed under function composition. It doesn't take too long to check that, but let's look at the details. Thus consider a pair

$$\mathfrak{C} \xrightarrow{\psi} \mathfrak{B} \xrightarrow{\phi} \mathfrak{A}$$

of limit arrows where

$$\mathfrak{C} = (\mathbb{C}, c, C, \mathcal{C}) \quad \mathfrak{B} = (\mathbb{B}, b, B, \mathcal{B}) \quad \mathfrak{A} = (\mathbb{A}, a, A, \mathcal{A})$$

are the three limit structures, and let $\theta = \phi \circ \psi$. We want $\theta \circ \mathcal{C} = \mathcal{A} \bullet \theta$, that is

$$(\theta \circ \mathcal{C})r = \mathcal{A}(\theta \circ r)$$

for each $r : \mathbb{N} \longrightarrow \mathbb{C}$. But³

$$\begin{aligned} (\theta \circ \mathcal{C})r &= (\phi \circ \psi \circ \mathcal{C})r \\ &= (\phi \circ (\mathcal{B} \bullet \psi))r \\ &= \phi((\mathcal{B} \bullet \psi)r) \\ &= \phi(\mathcal{B}(\psi \circ r)) \\ &= (\phi \circ \mathcal{B})(\psi \circ r) \\ &= (\mathcal{A} \bullet \phi)(\psi \circ r) \\ &= \mathcal{A}(\phi \circ \psi \circ r) = \mathcal{A}(\theta \circ r) \end{aligned}$$

as required.

The three categories are connected by a pair of forgetful functors

$$\mathbf{Limit} \xrightarrow{i} \mathbf{Peano} \xrightarrow{i} \mathbf{Set}$$

which we show have left adjoints. In particular, the composite functor

$$\mathbf{Limit} \xrightarrow{i} \mathbf{Set}$$

has a left adjoint, and so **Limit** has an initial object, the image of the empty set under the composite left adjoint.

In fact, the left adjoint of

$$\mathbf{Peano} \xrightarrow{i} \mathbf{Set}$$

is well known and easily described (although I can't remember ever having seen such a description).

²In fact, I do have a bit more information which I may write up at some point.

³In fact, this is easier if done directly without using \bullet .

Given an arbitrary set X we form a certain peano structure $\mathfrak{N}[X]$. This is carried by \mathbb{N} together with the set of all pairs

$$(m, x)$$

for $m \in \mathbb{N}$ and $x \in X$. The distinguished element is 0 (as an element of \mathbb{N}) and the successor operation S of $\mathfrak{N}[X]$ is given by

$$Sm = 1 + m \quad S(m, x) = (1 + m, x)$$

for all $m \in \mathbb{N}, x \in X$. Note that $\mathfrak{N}[\emptyset] = \mathfrak{N}$.

We take

$$\begin{array}{ccc} X & \xrightarrow{\eta} & \mathbb{N}[X] \\ x & \longmapsto & (0, x) \end{array}$$

as the unit arrow. This is Peano-epic, in the sense that for each pair of peano arrows

$$\mathfrak{N}[X] \begin{array}{c} \xrightarrow{\phi} \\ \xrightarrow{\psi} \end{array} \mathfrak{A}$$

to a peano structure $\mathfrak{A} = (\mathbb{A}, a, A)$, if $\phi \circ \eta = \psi \circ \eta$ then $\phi = \psi$. To see this observe that

$$\phi 0 = a = \psi 0 \quad \phi \circ S = A \circ \phi \quad \psi \circ S = A \circ \psi \quad \phi(0, x) = \psi(0, x)$$

for each $x \in X$, so that

$$\phi m = A^m a = \psi m$$

and

$$\phi(m, x) = (\phi \circ S^m)(0, x) = (A^m \circ \phi)(0, x) = (A^m \circ \psi)(0, x) = (\psi \circ S^m)(0, x) = \psi(m, x)$$

for each $m \in \mathbb{N}, x \in X$, as required.

With this we have the following result which shows that $\mathfrak{N}[X]$ is the reflection of X in Peano.

5.1 THEOREM. *For each peano structure \mathfrak{A} and function*

$$X \xrightarrow{\phi} \mathbb{A}$$

from a set to the carrier \mathbb{A} of \mathfrak{A} , there is a unique peano arrow

$$\mathfrak{N}[X] \xrightarrow{\phi^\#} \mathfrak{A}$$

such that

$$\begin{array}{ccc} X & \xrightarrow{\phi} & \mathbb{A} \\ & \searrow \eta & \nearrow \dot{i}(\phi^\#) \\ & \mathbb{N}[X] & \end{array}$$

commutes.

Proof. Since η is Peano-epic it suffices to exhibit just one such arrow ϕ^\sharp . To this end consider the function

$$\phi^\sharp : \mathbb{N}[X] \longrightarrow \mathbb{A}$$

given by

$$\phi^\sharp m = A^m a \quad \phi^\sharp(m, x) = A^m(\phi x)$$

for each $m \in \mathbb{N}, x \in X$. A couple of simple calculations leads to the required result. ■

The main purpose of this section is to show that the forgetful functor

$$\text{Limit} \xrightarrow{\iota} \text{Peano}$$

has a left adjoint. To do this we could try the same method we used for $\text{Peano} \longrightarrow \text{Set}$. We could look for an explicit description of the reflection of a peano structure in Limit . Unfortunately this requires a rather deeper knowledge of limit structures than we have at the moment. In particular, we don't even have an explicit description of the initial object \mathfrak{D} of Limit (for if we did then much of what is done here wouldn't be needed).

To those in the know, the existence of the reflector is immediate on general grounds, but mere existence is a bit unsatisfying. Eventually we want a structural analysis of Limit in general and \mathfrak{D} in particular. We are not yet in a position to produce such an analysis, but we can make a start.

To obtain the reflector we resort to a version of the adjoint functor theorem which, given suitable conditions, asserts the existence of a left adjoint. A discussion of this is given in pages 108 – 113 of [5]. We won't need the precise details of this result. Roughly speaking the suitable conditions come in two parts, a solution set condition and a requirement that certain limits behave in a nice way. In this section we give enough information to show that these conditions are met, but we won't spell out the details.

Let's first look at the solution set condition. This says that the limit structure freely generated from a peano structure doesn't grow out of control. We need to make that precise.

Each subobject of a limit structure $\mathfrak{A} = (\mathbb{A}, a, A, \mathcal{A})$ is determined by its carrier. These are precisely the subsets $\mathbb{B} \subseteq \mathbb{A}$ where $a \in \mathbb{B}$ with \mathbb{B} closed under A and \mathcal{A} . This last condition means that $\mathcal{A}p \in \mathbb{B}$ for each $p : \mathbb{N} \longrightarrow \mathbb{B}$. Almost trivially, the intersection of any family of subobjects of \mathfrak{A} is itself a subobject of \mathfrak{A} .

For each subset $X \subseteq \mathbb{A}$ the X -core of \mathfrak{A} is the intersection of all those subobjects which include X . We write $\mathfrak{A}(X)$ for this X -core. In particular, $\mathfrak{A}(\emptyset)$ is the unique smallest subobject of \mathfrak{A} , and is the range of the unique mediating arrow $\mathfrak{D} \longrightarrow \mathfrak{A}$ from the initial object (which we have yet to show exists).

How big is $\mathfrak{A}(X)$?

Given a set X let

$$|X| = \begin{cases} \text{cardinality of } X & \text{if this is infinite} \\ \aleph_0 & \text{otherwise} \end{cases}$$

so that $|X|$ is infinite. We use this to calculate an upper bound to the size of $\mathfrak{A}(X)$.

Let

$$AX = \{A^m x \mid m \in \mathbb{N}, x \in X \cup \{0\}\}$$

so that AX is the closure of $X \cup \{0\}$ under the successor of \mathfrak{A} . Clearly $AX \subseteq \mathfrak{A}(X)$, that is AX is a subset of the carrier of $\mathfrak{A}(X)$. With this let

$$X^+ = AX \cup \{\mathcal{A}p \mid p : \mathbb{N} \longrightarrow AX\}$$

so that $X \subseteq X^+ \subseteq \mathfrak{A}(X)$. The X -core of \mathfrak{A} is the least fixed point of this inflator. To find this we iterate $(\cdot)^+$. Thus we set

$$X^{(0)} = X \quad X^{(\alpha+1)} = X^{(\alpha)+} \quad X^{(\lambda)} = \bigcup \{X^{(\alpha)} \mid \alpha < \lambda\}$$

for each ordinal α and limit ordinal λ . On cardinality grounds this ascending chain of sets eventually stabilizes at $X^{(\infty)}$ for some ordinal ∞ , and then $X^{(\infty)}$ is the X -core.

Let Ω be the first uncountable ordinal (that is, ω_1). A cofinality argument shows that $X^{(\Omega)+} = X^{(\Omega)}$, and hence $\infty \leq \Omega$ in all cases. (For each limit ordinal λ we have $AX^{(\lambda)} \subseteq X^{(\lambda)}$. Consider any $p : \mathbb{N} \longrightarrow X^{(\Omega)}$. This gives an ω -chain through

$$X^{(\Omega)} = \bigcup \{X^{(\alpha)} \mid \alpha < \Omega\}$$

and hence there is some $\alpha < \Omega$ with $p[\mathbb{N}] \subseteq X^{(\alpha)}$. But now $\mathcal{A}p \in X^{(\alpha+1)} \subseteq X^{(\Omega)}$, as required.)

A simple calculation gives

$$|AX| = |X|$$

and hence

$$|X^+| \leq |X| + |AX|^{\aleph_0} = |X|^{\aleph_0}$$

holds. Using this a simple induction shows that

$$|X^{(\alpha)}| \leq |X|^{\aleph_0}$$

for all countable ordinals α , and hence

$$|\mathfrak{A}(X)| \leq |X^{(\Omega)}| \leq |X|^{\aleph_0}$$

to give an upper bound on the size of the X -core.

As a particular case of this we see that

$$|\mathfrak{D}| \leq |\emptyset|^{\aleph_0} = 2^{\aleph_0}$$

and it is not hard to show that this upper bound is achieved.

With this we can verify the solution set condition.

5.2 LEMMA. *For each peano structure \mathfrak{P} , there is a set I , an I -indexed family of limit structure \mathfrak{A}_i , and an I -indexed family of peano arrows*

$$\mathfrak{P} \xrightarrow{f_i} \mathfrak{A}_i$$

such that for each peano arrow

$$\mathfrak{P} \xrightarrow{f} \mathfrak{A}$$

to some limit structure \mathfrak{A} , there is at least one index i and a limit arrow

$$\mathfrak{A}_i \xrightarrow{\phi} \mathfrak{A}$$

giving a factorization

$$\begin{array}{ccc} \mathfrak{P} & \xrightarrow{f} & \iota\mathfrak{A} \\ & \searrow f_i & \nearrow \iota\phi \\ & \iota\mathfrak{A}_i & \end{array}$$

through f_i .

Proof. We are interested in those limit structures \mathfrak{B} with $|\mathfrak{B}| \leq |\mathfrak{P}|^{\aleph_0}$. Up to isomorphism, there is only a set of such structures. Consider any family of representatives of these isomorphism classes. For each such representative \mathfrak{B} there is only a set of peano arrows $\mathfrak{P} \longrightarrow \iota\mathfrak{B}$. In this way we produce the indexed set of arrows f_i .

Consider now any peano arrow

$$\mathfrak{P} \xrightarrow{f} \iota\mathfrak{A}$$

to some limit structure \mathfrak{A} . Let X be the range of this f , and consider the subobject $\mathfrak{A}(X)$ of \mathfrak{A} . Thus we may factorize f as

$$\mathfrak{P} \xrightarrow{g} \iota\mathfrak{A}(X) \xrightarrow{\iota} \iota\mathfrak{A}$$

a peano arrow followed by an insertion.

Since $|X| \leq |\mathfrak{P}|$ we have

$$|\mathfrak{A}(X)| \leq |X|^{\aleph_0} \leq |\mathfrak{P}|^{\aleph_0}$$

so that $\mathfrak{A}(X)$ is one of the structures \mathfrak{B} . Thus we may replace g by some f_i where $\mathfrak{A}(X)$ is isomorphic to \mathfrak{A}_i , and the required ϕ is the image of ι across this isomorphism. \blacksquare

This is the smallness condition required for the existence of a left adjoint. The other condition is concerned with the way limits behave in the two categories. Let's see how to calculate in **Limit** arbitrary products and arbitrary equalizers. (This will give us enough for an application of the adjoint functor theorem.)

We show that the category **Limit** has products, and these are essentially furnished cartesian products. To see this let

$$\mathbf{A} = (\mathfrak{A}_x \mid x \in X)$$

be an indexed family of limit structures

$$\mathfrak{A}_x = (\mathbb{A}_x, a_x, A_x, \mathcal{A}_x)$$

(for an arbitrary indexed set X). Consider the cartesian product

$$\mathbb{A}^\pi = \prod (\mathbb{A}_x \mid x \in X)$$

of the carrying set, Thus \mathbb{A}^π is the set of choice functions for $(\mathbb{A}_x \mid x \in X)$. For each $x \in X$ let

$$\text{proj}_x : \mathbb{A}^\pi \longrightarrow \mathbb{A}_x$$

be the projection function (given by ‘evaluation at x ’). We furnish \mathbb{A}^π as a limit structure \mathfrak{A}^π with each proj_x a limit arrow.

The distinguished element a^π is $(a_x \mid x \in X)$, the choice function that for each $x \in X$ selects a_x . The successor operation A^π is given by

$$A^\pi f x = A_x(fx)$$

(for $f \in \mathbb{A}^\pi$ and $x \in X$). Finally, the limit creator \mathcal{A}^π is given by

$$\mathcal{A}^\pi q x = \mathcal{A}_x(\text{proj}_x \circ q)$$

for $q : \mathbb{N} \longrightarrow \mathbb{A}^\pi$ and $x \in X$.

For each $x \in X$ and $f \in \mathbb{A}^\pi$ we have $\text{proj}_x f = fx$ and hence, the two properties

$$\text{proj}_x a^\pi = a_x \quad \text{proj}_x \circ A^\pi = A_x \circ \text{proj}_x$$

are immediate. Also, for $q : \mathbb{N} \longrightarrow \mathbb{A}^\pi$ we have

$$\mathcal{A}^\pi q x = \text{proj}_x(\mathcal{A}^\pi q) = (\text{proj}_x \circ \mathcal{A}^\pi)q \quad \mathcal{A}_x(\text{proj}_x \circ q) = (\mathcal{A}_x \bullet \text{proj}_x)q$$

so that the definition of \mathcal{A}^π can be rephrased as

$$\text{proj}_x \circ \mathcal{A}^\pi = \mathcal{A}_x \bullet \text{proj}_x$$

to verify that proj_x is a limit arrow $\mathfrak{A}^\pi \longrightarrow \mathfrak{A}_x$.

To show that \mathfrak{A}^π is the product of the family \mathbf{A} in Limit consider an X -indexed family

$$\mathfrak{B} \xrightarrow{\phi_x} \mathfrak{A}_x$$

of limit arrows with a common source \mathfrak{B} . We must show there is a unique mediating limit arrow

$$\mathfrak{B} \xrightarrow{\phi} \mathfrak{A}^\pi$$

such that $\phi_x = \text{proj}_x \circ \phi$ for each x .

Let

$$\mathfrak{B} = (\mathbb{B}, b, B, \mathcal{B})$$

(as usual). If there is a function

$$\phi : \mathbb{B} \longrightarrow \mathbb{A}^\pi$$

satisfying $\phi_x = \text{proj}_x \circ \phi$, for each $x \in X$, then we must have

$$\phi y x = (\text{proj}_x \circ \phi)y = \phi_x y$$

for each $y \in \mathbb{B}$. Thus it suffices to show that this ϕ is a limit arrow.

For each $x \in X$ we have

$$\phi b x = \phi_x b = a_x$$

so that $\phi b = a^\pi$. Also, for $y \in B, x \in X$ we have

$$\begin{aligned}
((\phi \circ B)y)x &= \phi(By)x \\
&= \phi_x(By) \\
&= (\phi_x \circ B)y \\
&= (A_x \circ \phi_x)y \\
&= A_x(\phi_x y) \\
&= A_x(\phi y x) \\
&= A^\pi(\phi y)x = ((A^\pi \circ \phi)y)x
\end{aligned}$$

so that $\phi \circ B = A^\pi \circ \phi$. It remains to check that

$$\phi \circ \mathcal{B} = \mathcal{A}^\pi \bullet \phi$$

to complete the proof. But, for each $q : \mathbb{N} \longrightarrow \mathbb{B}$ and $x \in X$ we have

$$\begin{aligned}
((\phi \circ \mathcal{B})q)x &= \phi(\mathcal{B}q)x \\
&= \phi_x(\mathcal{B}q) \\
&= (\phi_x \circ \mathcal{B})q \\
&= (\mathcal{A}_x \bullet \phi_x)q \\
&= \mathcal{A}_x(\phi_x \circ q) \\
&= \mathcal{A}_x(\text{proj}_x \circ \phi \circ q) \\
&= \mathcal{A}^\pi(\phi \circ q)x = ((\mathcal{A}^\pi \bullet \phi)q)x
\end{aligned}$$

to give the required result.

It is instructive to consider the case where \mathbf{A} is a constant family with

$$\mathfrak{A}_x = (\mathbb{A}, a, A, \mathcal{A})$$

for each $x \in X$. In this case \mathbb{A}^π is the function space $(X \longrightarrow \mathbb{A})$, and the constructed $\mathcal{A}^\pi \in \mathcal{L}(\mathbb{A}^\pi)$ satisfies

$$\mathcal{A}^\pi p x = \mathcal{A}(\text{proj}_x \circ p) = \mathcal{A}(u \mapsto p u x)$$

for each $p : \mathbb{N} \longrightarrow \mathbb{A}$ and $x \in \mathbb{A}$. In particular, when the index set X is also \mathbb{A} we obtain the lift \mathcal{A}' of \mathcal{A} as described in section 4.

We now turn to equalizers. Let Φ be a family of limit arrows

$$\mathfrak{B} \xrightarrow{\phi} \mathfrak{A}$$

with a common source and a common target. We require a limit arrow

$$\mathfrak{E} \xrightarrow{\epsilon} \mathfrak{A}$$

which universally equalizes the family Φ . We produce this by lifting the equalizer in **Set**. Let \mathbb{E} be the subset of \mathbb{B} on which all members of Φ agree. Thus

$$x \in \mathbb{E} \iff (\forall \phi, \psi \in \Phi)[\phi x = \psi x]$$

for $x \in \mathbb{B}$. We show that \mathbb{E} is closed under the attributes of \mathfrak{B} to produce \mathfrak{E} as a subobject of \mathfrak{B} .

Only the closure under the limit creator \mathcal{B} of \mathfrak{B} is not immediate. Consider any $p : \mathbb{N} \longrightarrow \mathbb{B}$ with range in \mathbb{E} . We want $\mathcal{B}p \in \mathbb{E}$. We have

$$\phi \circ p = \psi \circ p \quad \phi \circ \mathcal{B} = \mathcal{A} \bullet \phi \quad \psi \circ \mathcal{B} = \mathcal{A} \bullet \psi$$

for each $\phi, \psi \in \Phi$. Thus

$$\phi(\mathcal{B}p) = (\phi \circ \mathcal{B})p = (\mathcal{A} \bullet \phi)p = \mathcal{A}(\phi \circ p) = \mathcal{A}(\psi \circ p) = (\mathcal{A} \bullet \psi)p = (\psi \circ \mathcal{B})p = \psi(\mathcal{B}p)$$

as required.

This gives us the subobject \mathfrak{E} of \mathfrak{B} and, by construction, the insertion ϵ of \mathfrak{E} into \mathfrak{B} equalizes the family ϕ . We must check that ϵ universally equalizes Φ .

Consider any limit arrow

$$\mathfrak{C} \xrightarrow{\theta} \mathfrak{B}$$

where $\phi \circ \theta = \psi \circ \theta$ for all $\phi, \psi \in \Phi$. The range of θ is a subset of \mathbb{E} so that, as functions, $\theta = \epsilon \circ \delta$ for some unique function

$$\mathbb{C} \xrightarrow{\delta} \mathbb{E}$$

where \mathbb{C} is the carrier of \mathfrak{C} . It suffices to show that δ is a limit arrow, and again only the preservation of limit creators is not immediate. Thus we require

$$\delta \circ \mathcal{C} = \mathcal{E} \bullet \delta$$

where \mathcal{C}, \mathcal{E} are the limit creators of $\mathfrak{C}, \mathfrak{E}$. For each $p : \mathbb{N} \longrightarrow \mathbb{C}$ we have

$$\begin{aligned} \epsilon((\delta \circ \mathcal{C})p) &= (\epsilon \circ \delta \circ \mathcal{C})p \\ &= (\theta \circ \mathcal{C})p \\ &= (\mathcal{B} \bullet \theta)p \\ &= \mathcal{B}(\theta \circ p) \\ &= \mathcal{B}(\epsilon \circ \delta \circ p) \\ &= (\mathcal{B} \bullet \epsilon)(\delta \circ p) \\ &= (\epsilon \circ \mathcal{E})(\delta \circ p) \\ &= \epsilon(\mathcal{E}(\delta \circ p)) = \epsilon((\mathcal{E} \bullet \delta)p) \end{aligned}$$

and hence

$$(\delta \circ \mathcal{C})p = (\mathcal{E} \bullet \delta)p$$

since ϵ is injective, to give the required

$$\delta \circ \mathcal{C} = \mathcal{E} \bullet \delta$$

since p is arbitrary. In the block of equalities, the first is immediate, the second holds by the extraction of δ , the third holds since θ is a limit arrow, the fourth by the definition of ‘ \bullet ’, the fifth by the extraction of δ , the sixth by the definition of ‘ \bullet ’, the seventh since ϵ is a limit arrow, the eighth is immediate, the last by the definition of ‘ \bullet ’.

There is still much we don’t know about **Limit**, but this section gives us enough for the time being.

6 A system of informal notations

Each limit structure

$$\mathfrak{A} = (\mathbb{A}, a, A, \mathcal{A})$$

is a place where iterations can be carried out. Starting from the distinguished element a , we repeatedly apply the successor operation A , and every now and then we use the limit creator \mathcal{A} to collate a generated ω -chain of elements into a single element.

To carry out such an iteration we need a description of which ω -chains should be collated, and when. Of course, each element of such an ω -chain will be produced by a smaller iteration which may include some ω -chains and collations, and these may include deeper ω -chains, and so on. Thus the full description of an iteration could be a highly nested collection of collations and uses of the successor.

We claim that each term $\vdash \alpha : \mathcal{O}$ gives, or is, a template for such an iteration. Thus it should be possible to determine the element $\mathfrak{A}\langle\alpha\rangle$ of \mathfrak{A} produced by that iteration. As a particular case of this the term should name an ordinal $\langle\alpha\rangle \in \text{Ord}$ (obtained as $\mathfrak{D}\text{rd}\langle\alpha\rangle$). How do we produce these elements?

Using the semantics of $\lambda\mathbf{H}$ (as described in Section 3) each term $\vdash \alpha : \mathcal{O}$ gives an element $\llbracket\alpha\rrbracket \in \mathcal{O}$. Since \mathfrak{D} is the initial limit structure there is a unique limit arrow

$$\mathfrak{D} \xrightarrow{\mathfrak{A}(\cdot) = J_{\mathfrak{A}}} \mathfrak{A}$$

to the limit structure under consideration. Clearly $\mathfrak{A}\langle\alpha\rangle$ should be the image of $\llbracket\alpha\rrbracket$ across this arrow, that is

$$\mathfrak{A}\langle\alpha\rangle = \mathfrak{A}(\llbracket\alpha\rrbracket)$$

should hold.

One flaw with this method of producing $\mathfrak{A}\langle\alpha\rangle$ is that we don't know much about the structure \mathfrak{D} , and the function $\mathfrak{A}(\cdot)$ may be quite complicated even for limit structures \mathfrak{A} that we think we understand. Perhaps there is another method by which we move directly from the piece of syntax α to the element $\mathfrak{A}\langle\alpha\rangle$ without passing through \mathfrak{D} .

Much of the rest of this survey is concerned with developing such a method for a reasonably large class of terms. In particular, we show how to calculate the ordinal $\mathfrak{D}\text{rd}\langle\alpha\rangle$ named by such a term α .

The source of the difficulties is that calculating $\mathfrak{A}\langle\alpha\rangle$ directly from the derivation of α will involve higher order gadgets. We need some way of keeping these excursions to a minimum.

In this section we do not work directly in the language of $\lambda\mathbf{H}$. As a preliminary skirmish we set up a small auxiliary language \mathbf{IN} of terms we call *informal notations* (even though they are quite formal). Each $\alpha \in \mathbf{IN}$ is a rather simple piece of syntax, and the intention is that it describes a reasonably simple iteration template. Accordingly, for each $\alpha \in \mathbf{IN}$ and each limit structure \mathfrak{A} we generate an element $\mathfrak{A}\llbracket\alpha\rrbracket$ of \mathfrak{A} , the meaning of α in \mathfrak{A} . In particular, each $\alpha \in \mathbf{IN}$ has a meaning $\llbracket\alpha\rrbracket = \mathfrak{D}\llbracket\alpha\rrbracket$, and we find that

$$\mathfrak{A}\llbracket\alpha\rrbracket = \mathfrak{A}(\llbracket\alpha\rrbracket)$$

using the mediating arrow $\mathfrak{A}(\cdot)$ from \mathfrak{D} to \mathfrak{A} .

Eventually we want a similar method for term $\vdash \alpha : \mathcal{O}$ of $\lambda\mathbf{H}$, so we may think of this section as a test run using uncomplicated terms. Later, in Section 7 we show how

each $\alpha \in \mathbf{IN}$ can be simulated in $\lambda\mathbf{G}$, and then in Section 10 we show how each such α has a formal analogue in $\lambda\mathbf{H}$.

The calculations of Section 5 show that \mathbb{O} is large and rather complex. Thus we can not expect to be able to name each of its members using simple methods. The auxiliary language \mathbf{IN} will get at just a small part of \mathbb{O} . How can we try to name members of \mathbb{O} ? The use of 0 (the base element) and S (the successor operation) is no problem, but the use of the *Lim* construction is. This is because it necessarily invokes higher order gadgets. To keep these to a minimum we introduce a couple more low level constructs which will take the place of certain *Lim* constructs. Of course, we can not remove all uses of *Lim*, so we allow just one explicit use.

If, for a moment, we think in terms of ordinals, then the informal notations are concerned with the ordinals below ϵ_0 described in cantor normal form to base ω . In fact, syntactically there is no difference between these descriptions of ordinals and the informal notation, but the meanings are different.

6.1 DEFINITION. The family \mathbf{IN} of informal notations is generated as follows.

- There are two primitive notations, 0 and ω .
- If α, η, ζ are notations then so are $S\alpha, (\zeta + \eta)$, and ω^α .

There are no other members of \mathbf{IN} . ■

The syntax is chosen to be suggestive, to remind us of certain ordinals. However, we must remember that these notations describe iterations not ordinals, so there are some crucial differences in meaning. For instance $(\zeta + \eta)$ describes the iteration ζ followed by the iteration η . Thus

$$\omega + \omega^\omega \quad \omega^\omega$$

are different iterations whereas they are the same ordinal. (Different iterations when applied in a particular case might give the same result.)

How do we give $\alpha \in \mathbf{IN}$ a meaning $\mathfrak{A}[\alpha]$ relative to a limit structure \mathfrak{A} ? Before we can do this we need a global semantic input. We need to say what the primitive symbol ω should mean.

Think again of ordinals as iteration templates. In this case the meaning of ω is relative to some chosen fundamental sequences, and this choice permeates through the meanings of the other ordinal notations. We do a similar thing here.

6.2 CONVENTION. Let

$$\varpi[\cdot] : \mathbb{N} \longrightarrow \mathbb{N}$$

be an arbitrary function. The meaning of each $\alpha \in \mathbf{IN}$ is relative to $\varpi[\cdot]$. ■

We think of $\varpi[\cdot]$ as the analogue of the chosen fundamental sequence of ω . When we deal with ordinals we need some restrictions on $\varpi[\cdot]$ (such as being strictly monotone) but for iterations we can choose any function whatsoever.⁴

Each limit ordinal (below ϵ_0) needs a fundamental sequence. It is worth remembering how these can be generated from $\varpi[\cdot]$. If we take a standard route then we may set

$$\omega^{\alpha+1}[u] = \omega^\alpha \cdot \varpi[u] \quad \omega^\mu[u] = \omega^{\mu[u]} \quad (\zeta + \eta)[u] = \zeta + \eta[u]$$

⁴I do know that ϖ is a variant of π and not of ω . I've just typed the damn thing haven't I.

for each appropriate α, μ, ζ, η and $u \in \mathbb{N}$. There are other non-standard (even eccentric) ways of generating fundamental sequences, but we need not worry about those here. It is the selection of the fundamental sequences which partly determines the complexity of the iteration behaviour of the ordinal.

Fix the function $\varpi[\cdot]$, and consider a limit structure $\mathfrak{A} = (\mathbb{A}, a, A, \mathcal{A})$. We generate the meaning

$$\mathfrak{A}[\alpha] = J_{\mathbb{A}}\alpha\mathcal{A}Aa$$

of $\alpha \in \mathbf{IN}$ by recursion over its structure (as a piece of syntax). For this it is more convenient to work in terms of $J_{\mathbb{A}}$.

6.3 DEFINITION. For each limit structure $\mathfrak{A} = (\mathbb{A}, a, A, \mathcal{A})$ the element

$$J_{\mathbb{A}}\alpha\mathcal{A}Aa$$

is generated by recursion over $\alpha \in \mathbf{IN}$ as follows.

$$\begin{aligned} \text{(Zero)} \quad & J_{\mathbb{A}}0\mathcal{A}Aa = a \\ \text{(Step)} \quad & J_{\mathbb{A}}(S\alpha)\mathcal{A}Aa = A(J_{\mathbb{A}}\alpha\mathcal{A}Aa) \\ \text{(Omega)} \quad & J_{\mathbb{A}}\omega\mathcal{A}Aa = \mathcal{A}(u \mapsto A^{\varpi[u]}a) \\ \text{(Sum)} \quad & J_{\mathbb{A}}(\zeta + \eta)\mathcal{A}A = (J_{\mathbb{A}}\eta\mathcal{A}A) \circ (J_{\mathbb{A}}\zeta\mathcal{A}A) \\ \text{(Exp)} \quad & J_{\mathbb{A}}(\omega^\alpha)\mathcal{A} = (J_{\mathbb{A}}\alpha\mathcal{A}')(J_{\mathbb{A}}\omega\mathcal{A}) \end{aligned}$$

In the (Omega) clause the selected function $\varpi[\cdot]$ and the limit creator \mathcal{A} are used. In the (Exp) clause the lifting process is used. ■

What is the idea behind these clauses? As we said above, part of the idea is to hide as much as possible the uses of \mathcal{A} . Only in the (Omega) clause is there an explicit call on this. However, it will be brought to the surface as we unravel the uses of the other clauses. The idea of the (Sum) clause is to follow one iteration by another. At this stage the idea behind the (Exp) clause is harder to describe, but it will become clearer later. Notice that it encodes an iteration of an iteration gadget, but that doesn't explain why ' ω^α ' is the appropriate notation.

By taking the initial limit structure $\mathfrak{D} = (\mathbb{O}, 0, S, Lim)$ we see that each $\alpha \in \mathbf{IN}$ names an element $[\alpha] \in \mathbb{O}$. With a little care we may let ' α ' be both the informal notation and the member of \mathbb{O} . The syntax ' 0 ' and ' S ' has been chosen with this in mind. For $\alpha, \beta \in \mathbf{IN}$ we may write

$$\alpha = \beta$$

for the more cumbersome

$$[\alpha] = [\beta] \quad \mathfrak{D}[\alpha] = \mathfrak{D}[\beta] \quad J_{\mathbb{O}}\alpha Lim S 0 = J_{\mathbb{O}}\beta Lim S 0$$

all of which mean the same thing. We will develop a small system of rules for verifying such equalities. It is reasonable to expect that this will look something like the arithmetic of ordinals but with fewer identities.

By working through the various clauses we find that

$$\zeta + 0 = \zeta \quad \zeta + S\alpha = S(\zeta + \alpha) \quad \zeta + Lim p = Lim(u \mapsto \zeta + pu)$$

hold (as members of \mathbb{O}). This is just the recursive specification of ordinal addition lifted to \mathbb{O} . We needn't verify these clauses for we are going to do a similar thing with the more complicated Exp construction, and we will look at those details.

We need a little bit of preparation.

For each $m \in \mathbb{N}$ we may safely confuse m with $S^m 0$ (both as a member of \mathbf{IN} and a member of \mathbb{O}). By repeated use of the (Step) clause we find that

$$J_{\mathbb{A}} m \mathcal{A} A a = A^m a$$

holds for each limit structure \mathfrak{A} . For each $\mu \in \mathbf{IN}$ and $m \in \mathbb{N}$ let

$$\mu \cdot m \quad \text{abbreviate} \quad \mu + \cdot + \mu$$

where there are m copies of μ and brackets may be inserted to punctuate the sum. Using the (Sum) clause of Definition 6.3 we see that

$$J_{\mathbb{A}}(\mu \cdot m) \mathcal{A} A a = (J_{\mathbb{A}} \mu \mathcal{A} A)^m a$$

holds. (We could develop a notion of the product of two members of \mathbf{IN} , and then $\mu \cdot m$ will be such a product. We don't need to do this just yet, but will do so in Section 9.)

It turns out that

$$\omega^0 = S0 \quad \omega^{S\alpha} = \text{Lim}(u \mapsto \omega^\alpha \cdot \varpi[u]) \quad \omega^{\text{Lim } p} = \text{Lim}(u \mapsto \omega^{pu})$$

hold where the middle clause uses the abbreviation describe above. This is just the recursive specification of ordinal exponentiation lifted to \mathbb{O} . Let's look at the details of the proofs of the second and third clauses.

Notice that the (Omega) clause of Definition 6.3 gives

$$\omega = \llbracket \omega \rrbracket = J_{\mathbb{O}} \text{Lim} S0 = \text{Lim}(u \mapsto S^{\varpi[u]} 0) = \text{Lim} \varpi[\cdot]$$

as we might expect.

For the second identity we have

$$\begin{aligned} \omega^{S\alpha} &= J_{\mathbb{O}}(\omega^{S\alpha}) \text{Lim} S0 \\ &= (J_{\mathbb{O}'}(S\alpha) \text{Lim}') (J_{\mathbb{O}} \omega \text{Lim}) S0 \\ &= (J_{\mathbb{O}} \omega \text{Lim}) ((J_{\mathbb{O}'} \alpha \text{Lim}') (J_{\mathbb{O}} \omega \text{Lim}) S) 0 \\ &= (J_{\mathbb{O}} \omega \text{Lim}) (J_{\mathbb{O}}(\omega^\alpha) \text{Lim} S) 0 = \text{Lim } p \end{aligned}$$

where

$$pu = (J_{\mathbb{O}} \varpi[u] \text{Lim}) (J_{\mathbb{O}}(\omega^\alpha) \text{Lim} S) 0 = (J_{\mathbb{O}}(\omega^\alpha) \text{Lim} S)^{\varpi[u]} 0 = J_{\mathbb{O}}(\omega^\alpha \cdot \varpi[u]) \text{Lim} S0$$

to give the required result. In the first block of equalities, the first is the definition of $\omega = \llbracket \omega \rrbracket$, the second uses the (Exp) clause, the third uses the (Step) clause, and the fourth is a second use of the (Exp) clause. In the second block of equalities, the first uses the (Leap) clause of the specification of $J_{\mathbb{O}'} \text{rd}$ and $\omega = \text{Lim} \varpi[\cdot]$ from above, the fourth follows since $\varpi[u]$ is a natural number, and the third uses the observation concerning $\mu \cdot m$ above.

For the third identity we have

$$\omega^{\text{Lim } p} = J_{\mathbb{O}}(\omega^{\text{Lim } p}) \text{Lim} S0 = (J_{\mathbb{O}'}(\text{Lim } p) \text{Lim}') (J_{\mathbb{O}} \omega \text{Lim}) S0 = \text{Lim}' P0 = \text{Lim } q$$

where, for $u \in \mathbb{N}$

$$Pu = (J_{\mathbb{O}'}(pu)Lim')(J_{\mathbb{O}}\omega Lim)S = J_{\mathbb{O}}(\omega^{pu})LimS$$

and

$$qu = Pu0 = J_{\mathbb{O}}(\omega^{pu})LimS0$$

to give the required result. In the first block of equalities, the first is the definition of $\omega^{Lim p} = \llbracket \omega^{Lim p} \rrbracket$, the second uses the (Exp) clause, and the third and fourth depend on the definitions of P and q . In the second block of equalities, the first is the definition of P using the (Leap) clause of the specification of $J_{\mathbb{O}'}$, and the second uses the (Exp) clause. In the third block of equalities, the first is the definition of q (from the first block) remembering how limit creators are lifted, and the second equality makes use of the second block.

This is not very exciting is it, but at least it's a start. In the next section we show how these informal notations can be simulated in $\lambda\mathbf{G}$. In a sense they exhaust the power of $\lambda\mathbf{G}$, but that is not proved here. Later, in Section 10, we will show that these informal notations can be immersed directly in $\lambda\mathbf{H}$, not just simulated. We then find there are many, many more iterations templates available in $\lambda\mathbf{H}$.

7 Naming in $\lambda\mathbf{G}$

The primary job of $\lambda\mathbf{G}$ is to name numeric gadgets. For each $m \in \mathbb{N}$ we set

$$\underline{m} = \underline{S}^m \underline{0}$$

to obtain the numeral for m , the canonical name for m in $\lambda\mathbf{G}$. Setting

$$\begin{aligned} \underline{A} &= \lambda v, u : \mathcal{N} . \mathbb{I}_{\mathcal{N}} u \underline{S} v \\ \underline{M} &= \lambda v, u : \mathcal{N} . \mathbb{I}_{\mathcal{N}} u (\widehat{A}v) \underline{0} \quad \text{where} \quad \widehat{A}v = \lambda w : \mathcal{N} . \underline{A} w v \\ \underline{E} &= \lambda v, u : \mathcal{N} . \mathbb{I}_{\mathcal{N}} u (\widehat{M}v) \bar{1} \quad \text{where} \quad \widehat{M}v = \lambda w : \mathcal{N} . \underline{M} w v \end{aligned}$$

we obtain names for addition, multiplication, and exponentiation. We may check this by showing that

$$\underline{A} \underline{n} \underline{m} \triangleright \underline{n + m} \quad \underline{M} \underline{n} \underline{m} \triangleright \underline{n \times m} \quad \underline{E} \underline{n} \underline{m} \triangleright \underline{n^m}$$

hold for all $m, n \in \mathbb{N}$. The terms $\underline{M}, \underline{E}$ use subterms $\widehat{A}v, \widehat{M}v$ to swap the order of the two arguments. This could be avoided if we let $\underline{M}, \underline{E}$ consume the arguments in a different order. However, the order used here simplifies some of the constructions considered later.

Many more numeric functions can be named in $\lambda\mathbf{G}$. For instance, the term

$$\underline{B} = \lambda w, v, u : \mathcal{N} . \mathbb{I}_{\mathcal{N}} u (\underline{E}v) w$$

names the stacking function \beth where

$$\beth(k, n, 0) = k \quad \beth(k, n, m + 1) = n^{\beth(k, n, m)}$$

for $m, n, k \in \mathbb{N}$.

We need not go into the details of what can be done in $\lambda\mathbf{G}$. Much of this can be found in [25]. However, a few words will help to motivate what follows.

7.1 **EXAMPLE.** Consider the long Grzegorzcyk hierarchy developed in Examples 3.4(b) and 3.6. Let S be the base function and let **ack** be the jump functional (and use the diagonal limit creator Δ). This gives us a function

$$\alpha \longmapsto \mathbb{G}\alpha\mathbf{ack}S$$

from some suitable collection of ordinal notations α to \mathbb{N}' . This is often called the fast-growing hierarchy.

To be precise let us select

$$\omega[\cdot] = S$$

as the fundamental sequence for ω and use the standard method (as explained just after Convention 6.2) to generate the fundamental sequence for each ordinal limit below ϵ_0 . Also set

$$\epsilon[0] = \omega \quad \epsilon[r+1] = \omega^{\epsilon[r]}$$

(for $r \in \mathbb{N}$) to produce a standard fundamental sequence for ϵ_0 itself. With these choices we obtain an ω -chain

$$r \longmapsto \mathbb{G}\epsilon[r]\mathbf{ack}S \quad (r \in \mathbb{N})$$

of functions in \mathbb{N}' . Each of the functions $\mathbb{G}\epsilon[r]\mathbf{ack}S$ can be named in $\lambda\mathbf{G}$ (but the whole 2-placed function can't).

To see this set

$$\mathbf{ack}_s = \lambda x_{s+1} : \mathcal{N}^{(s+1)}, x_s : \mathcal{N}^{(s)}, \dots, x_1 : \mathcal{N}', x_0 : \mathcal{N} . \mathbb{I}_{\mathcal{N}^{(s)}}(\underline{S}x_0)x_{s+1}x_s \cdots x_1x_0$$

to obtain a term

$$\vdash \mathbf{ack}_s : \mathcal{N}^{(s+2)}$$

of $\lambda\mathbf{G}$. It isn't too hard to show that \mathbf{ack}_0 names **ack**, but it take a bit more effort to show that

$$\mathbf{ack}_{r+1}\mathbf{ack}_r \cdots \mathbf{ack}_1\mathbf{ack}_0 \underline{S}$$

names $\mathbb{G}\epsilon[r]\mathbf{ack}S$. ■

This example is a path through the limit structure

$$(\mathbb{N}', f, F, \Delta)$$

for $f = S$ and $F = \mathbf{Ack}$. By choosing other f and F that can be named in $\lambda\mathbf{G}$ (and even modifying the limit creator) we can name many more functions *provided we can simulate sufficiently many the ordinals* in $\lambda\mathbf{G}$. Showing how to do that is the topic of this section.

In section 6 we described a system \mathbf{IN} of informal notation. Given some function $\varpi[\cdot] : \mathbb{N}'$ these name certain of the 'smaller' members of \mathbb{O} , and when viewed as ordinals they form a system of ordinal notations generated using $\varpi[\cdot]$ as the fundamental sequence for ω . We now show how these notation can be simulated in $\lambda\mathbf{G}$.

The technique is well know. Given a type σ with interpretation $\mathbb{A} = \llbracket \sigma \rrbracket$, we try to view each informal notation as an iteration gadget on \mathbb{A} . In other words, each $\alpha \in \mathbf{IN}$ has a whole family of simulating term α_σ in $\lambda\mathbf{G}$ indexed by the types.

This idea is used in [25], where it is described in terms of ordinals, but easily extends to all iteration templates. (When writing [25] I had this in mind but didn't want to clutter up the book with an explicit development.)

In [25] Definition 8.9 (on page 184) is essentially that of an informal notation. (The phrase ‘canonical notation’ as used there should not be confused with the same phrase as used here.) That definition is couched in terms of ordinals, but that is merely camouflage. A simulation of these terms is given by Definition 9.22 (on on page 203). We use the same method here.

For each type σ we translate each $\alpha \in \mathbf{IN}$ into a term

$$\alpha_\sigma = \lambda l : \mathcal{L}(\sigma), y : \sigma', x; \sigma . \text{BODY}_\sigma(l, y, x)$$

where $\vdash \alpha_\sigma : \mathcal{L}(\sigma) \rightarrow \sigma''$ holds. It is the construction of the body that is important. This is done by recursion over the structure of α (as a piece of syntax) with variation of the parameter σ . This parameter variation is the crucial trick. Since the prefix is the same throughout we let

$$\lambda_\sigma l, y, x \text{ abbreviate } \lambda l : \mathcal{L}(\sigma), y : \sigma', x : \sigma$$

so that

$$\alpha_\sigma = \lambda_\sigma l, y, x . \text{BODY}_\sigma(l, y, x)$$

is the term we must generate.

As well as the minor step from ordinal to iteration template, there is a further complication we must handle. Here the meaning of each $\alpha \in \mathbf{IN}$ is relative to a selected function $\varpi[\cdot]$. In [25] the ordinal notations are generated from the particular case $\varpi[\cdot] = id_{\mathbb{N}}$. To deal with the more general case we assume given some term

$$\vdash \varpi : \mathcal{N}'$$

and generate the terms α_σ relative to this term. Thus we assume that the function $\varpi[\cdot]$ is named by the term ϖ . (This, of course, put a restriction on the complexity of $\varpi[\cdot]$.)

7.2 DEFINITION. For each $\alpha \in \mathbf{IN}$ and type σ the $\lambda\mathbf{G}$ -term α_σ is obtained by recursion over the structure of α using the following clauses.

$$\begin{aligned} \text{(Zero)} \quad & 0_\sigma = \lambda_\sigma l, y, x . x \\ \text{(Step)} \quad & (S\alpha)_\sigma = \lambda_\sigma l, y, x . y(\alpha_\sigma l y x) \\ \text{(Omega)} \quad & \omega_\sigma = \lambda_\sigma l, y, x . l(\lambda u : \mathcal{N}' . l_\sigma(\varpi u) y x) \\ \text{(Sum)} \quad & (\zeta + \eta)_\sigma = \lambda_\sigma l, y, x . (\eta_\sigma l y)(\zeta_\sigma l y x) \\ \text{(Exp)} \quad & (\omega^\alpha)_\sigma = \lambda_\sigma l, y, x . (\alpha_{\sigma'} l)(\omega_\sigma l) y x \end{aligned}$$

In the (Omega) clause the term ϖ naming the selected function $\varpi[\cdot]$ is used. In the (Exp) clause the formal lifting process is used. ■

An easy induction shows that

$$\vdash \alpha_\sigma : \mathcal{L}(\sigma) \rightarrow \sigma''$$

for each $\alpha \in \mathbf{IN}$. In the same way it is routine to verify the analogue of Theorem 9.24 of [25] (on page 204).

7.3 THEOREM. *Suppose $\llbracket \sigma \rrbracket = \mathbb{A}$, then for each $\alpha \in \mathbf{IN}$ the term α_σ names $J_{\mathbb{A}}\alpha$.*

Of course, these informal notations don't get very far. As ordinals they stay below ϵ_0 . We want to demonstrate how $\lambda\mathbf{H}$ can get much further than that. This will take quite a bit of preparation in which we seem to stay below ϵ_0 , but eventually towards the end of Section 10 we will start to take off.

8 Equational reasoning and canonical notations

Different terms

$$\vdash \alpha : \mathcal{O} \quad \vdash \beta : \mathcal{O}$$

can have the same meaning $\llbracket \alpha \rrbracket = \mathfrak{D}(\alpha) = \mathfrak{D}(\beta) = \llbracket \beta \rrbracket$. The syntactic difference may be superficial, such as a different choice of bound identifiers; or it may be more substantial, such as a reduction equivalence between the terms. It seems a sensible move that when calculating $\llbracket \alpha \rrbracket$ we should first massage α into a more amenable form. Precisely what this ‘amenable form’ could be is unclear, but we should be able to make some sense of it.

We need some method of converting a term α into an ‘equivalent’ term α' . Clearly the reduction mechanism has a role to play in the conversion $\alpha \mapsto \alpha'$, but we need more. We need some **equational reasoning**. Furthermore, this reasoning can not be restricted just to inhabitants of \mathcal{O} .

We need a method by which when supplied with two judgements

$$\Gamma \vdash t_1 : \tau \quad \Gamma \vdash t_2 : \tau$$

(in the same context and with the same predicate) we can demonstrate (syntactically) that t_1 and t_2 have the same meaning (assuming that they do). In short, we want to be able to show that t_1 and t_2 are equal. Of course, we are not asking for a complete method, one that will demonstrate all possible equalities. We require only that the method is sound and will demonstrate a sufficient number of equalities.

To do this we extend the syntactic mechanisms of **λH** . First of all we introduce a new syntactic category

$$\Gamma \vdash [t_1 \approx t_2] : \tau$$

each member of which is an **Equational-judgement** (or E-judgement for short). This is formed from a context Γ , two terms t_1, t_2 , a type τ , a new symbol \approx , and some punctuation indicated by square brackets. We also set up a system of **E-derivations** which will witness the correctness of certain E-judgements. This new system must have certain properties. Firstly, there is a soundness requirement. If an E-judgements, as above, is deemed correct by the system then the two judgements

$$\Gamma \vdash t_1 : \tau \quad \Gamma \vdash t_2 : \tau$$

must be derivable (in the system of Section 2) and the interpretations of these two terms (derivations) must be the same. Secondly, there is a partial adequacy requirement. For a sufficiently large collection of pairs of judgements, as above, if these have the same meaning, then there is an E-derivation which demonstrates this.

What about the actual methods used? An E-derivations is a tree of E-judgements augmented by certain (typing) judgements and reductions, and built in accordance with certain rules. There are some obvious rules which will form part of any system of equational reasoning and which we need not set down here. A full list is given in Table 3 of [λH]. However, there are two rules which we must pay particular attention to.

Let \bowtie be the reflexive, symmetric, transitive closure of the reduction relation \triangleright (or the 1-step relation \triangleright). Thus $t_1 \bowtie t_2$ holds if we can move from t_1 to t_2 via a sequence of reductions, some going up and some going down. (If we invoke the confluence property

then we need no more than one reduction in either direction.) From the remarks above we see that

$$\frac{\Gamma \vdash t_1 : \tau \quad \Gamma \vdash t_2 : \tau \quad t_1 \bowtie t_2}{\Gamma \vdash [t_1 \approx t_2] : \tau}$$

should be a rule of equational reasoning.

There is also a more important rule.

8.1 DEFINITION. (The ω -rule or external induction principle) Suppose both

$$\Gamma \vdash \phi : (\mathcal{N} \rightarrow \tau) \quad \Gamma \vdash \psi : (\mathcal{N} \rightarrow \tau)$$

are derivable. Suppose also that for each $m \in \mathbb{N}$ the E-judgement, as to the left

$$\Gamma \vdash [\phi \underline{m} \approx \psi \underline{m}] : \tau \quad \Gamma \vdash [\phi \approx \psi] : (\mathcal{N} \rightarrow \tau)$$

is derivable. Then the E-judgement to the right is allowed. ■

This is an infinitary rule of E-derivation

$$\frac{\Gamma \vdash [\phi \underline{0} \approx \psi \underline{0}] : \tau \quad \Gamma \vdash [\phi \underline{1} \approx \psi \underline{1}] : \tau \quad \dots \quad \Gamma \vdash [\phi \underline{m} \approx \psi \underline{m}] : \tau \quad \dots}{\Gamma \vdash [\phi \approx \psi] : (\mathcal{N} \rightarrow \tau)} \quad (m \in \mathbb{N})$$

which requires an ω -chain of numerators to produce the denominator. What should we make of it?

To use the rule we need an ω -chain of smaller E-derivations. Since we can not possibly write down each one separately, we will be forced to describe some scheme which generates them. This will impose some uniformity on the numerator derivations. The nature of that uniformity must have something to do with the complexity of the gadget named by the terms ϕ and ψ .

At some later time (but not here) we will want to immerse this system of equational reasoning into some finitary system. This will enable certain uses of the ω -rule to be simulated by finite constructions, and again this will impose some uniformity on the uses. However, the extended system will not be able to handle other instances of the ω -rule. Thus, such enrichments are one way of measuring the complexity of instances of the ω -rule, and hence give an indication of the complexity of the concrete gadgets named by the system. Howard's original system and Feferman's system are two examples of such enrichments.

Here we will stick to the informal rule, and observe that all the instances we use are rather straight forward. In the next section we will see several such instances, but for now let's return to the evaluation problem.

Given a limit structure \mathfrak{A} and a term $\vdash \alpha : \mathcal{O}$, how do we calculate the element $\mathfrak{A}(\alpha)$ of \mathfrak{A} ? Notice that now we do not differentiate between the piece of syntax α and the concrete object $[[\alpha]] \in \mathcal{O}$. This will not lead to confusion here.

We can certainly calculate $\mathfrak{A}(\alpha)$ provided α is in an appropriately simple form.

8.2 DEFINITION. The algebraic terms are generated recursively as follows.

(Base) $\bar{0}$ is an algebraic term.

(Step) If α is an algebraic term then so is $\bar{S}\alpha$.

(Leap) If $\vdash p : (\mathcal{N} \rightarrow \mathcal{O})$ is derivable and for each $m \in \mathbb{N}$ there is an algebraic term α_m with $\vdash [pm \approx \alpha_m] : \mathcal{O}$, then Limp is an algebraic term.

In particular, $\vdash \alpha : \mathcal{O}$ holds for each algebraic term α .

A term $\vdash \alpha : \mathcal{O}$ is **canonical** (on level 0) if there is an algebraic term β such that $\vdash [\alpha \approx \beta] : \mathcal{O}$ is derivable. ■

For an algebraic term α the element $\mathfrak{A}(\alpha)$ is produced by simply unravelling the construction of α and applying the appropriate recursion clause at each stage. For instance, suppose $\alpha = \text{Limp}$ where for each $m \in \mathbb{N}$ we have $\vdash [pm \approx \alpha_m] : \mathcal{O}$ with α_m algebraic. By recursion, for each such m we have an element $\mathfrak{A}(\alpha_m) \in \mathbb{A}$, so we set $\mathfrak{A}(\alpha) = \mathcal{A}(m \mapsto \mathfrak{A}(\alpha_m))$

For each canonical term α we have an algebraic term β with $\vdash [\alpha \approx \beta] : \mathcal{O}$, and then we set $\mathfrak{A}(\alpha) = \mathfrak{A}(\beta)$. There is, of course, a minor problem here. Suppose

$$\vdash [\alpha \approx \beta] : \mathcal{O} \quad \vdash [\alpha \approx \gamma] : \mathcal{O}$$

for different algebraic terms β, γ . Which one do we use? Under such circumstances we have

$$\vdash [\beta \approx \gamma] : \mathcal{O}$$

and then β, γ will name the same element of Ord , so it shouldn't matter which one we use. In fact, it seems reasonable to conjecture that β and γ are essentially the same terms, that is they are alphabetic variants and differ only in the choice of bound variables. Let's not worry about that.

(If you must worry about it, then you can simply predicate the calculation of $\mathfrak{A}(\alpha)$ on the choice of the E-derivations used to move from α to the β with $\vdash [\alpha \approx \beta] : \mathcal{O}$.)

Almost trivially for each $m \in \mathbb{N}$ the term $\bar{m} = \bar{S}^m \bar{0}$ is algebraic. In the next section we will exhibit many more examples of canonical terms. In particular, we will see that all the 'obvious' names of ordinals are canonical.

In the remainder of this section we develop a method of showing terms are canonical. We use higher level canonicity.

8.3 DEFINITION. For each $k < \omega$ the class \mathcal{C}_k of terms is generated by recursion on k .

(0) The class \mathcal{C}_0 is exactly the class of canonical terms (on level 0).

(k+1) A term C is in \mathcal{C}_{k+1} if $\vdash C : \mathcal{O}^{(k+1)}$ and $CB \in \mathcal{C}_k$ for each $B \in \mathcal{C}_k$.

We say a term is **canonical** (on level k) if it is \mathcal{C}_k . ■

Notice that $C \in \mathcal{C}_{k+1}$ exactly when $CA_k \cdots C_0 \in \mathcal{C}_0$ for each $A_k \in \mathcal{C}_k, \dots, A_0 \in \mathcal{C}_0$.

An easy exercise shows that each class \mathcal{C}_k is closed under \approx , that is $C \in \mathcal{C}_k$ whenever $\vdash C : \mathcal{O}^{(k)}$ and $\vdash [C \approx B] : \mathcal{O}^{(k)}$ for some $B \in \mathcal{C}_k$.

We need some examples of terms in these classes. Trivially we have $\bar{0} \in \mathcal{C}_0$ and $\bar{S} \in \mathcal{C}_1$, which gives $\bar{m} \in \mathcal{C}_0$. Shortly we will see how the iterators give members of \mathcal{C}_{k+2} for each $k < \omega$.

Let's introduce some simplifying notation.

Starting from the atom \mathcal{O} we may generate a sequence $(\mathcal{O}^{(l)} \mid l < \omega)$ of types by

$$\mathcal{O}^{(0)} = \mathcal{O} \quad \mathcal{O}^{(r+1)} = \mathcal{O}^{(r)'}$$

for each $r \in \mathbb{N}$. We write $J^{(l)}$ for J_σ where σ is $\mathcal{O}^{(l)}$. In particular, we write J for $J^{(0)} = J_{\mathcal{O}}$. This will save a bit of clutter. We do a similar thing with limit lifting.

In Definition 4.4 we attached a term

$$\vdash \uparrow_\sigma : \mathcal{L}(\sigma) \rightarrow \mathcal{L}(\sigma')$$

to each type σ . This named the concrete limit lifter associated with $\mathbb{A}[\sigma]$. Using these terms \uparrow_σ and the constant Lim we generate terms $\mathbf{L}^{(l)}$ by

$$\mathbf{L}^{(0)} = \text{Lim} \quad \mathbf{L}^{(r+1)} = \uparrow_\sigma \mathbf{L}^{(r)} \quad \text{where } \sigma = \mathcal{O}^{(r)}$$

(for $r \in \mathbb{N}$), and we let $\mathbf{L} = \mathbf{L}^{(0)} = \text{Lim}$. As with concrete limit lifting, this notation omits a lot of information. It can be checked that

$$\vdash J^{(l)} : \mathcal{O} \rightarrow \mathcal{L}(\mathcal{O}^{(l)}) \rightarrow \mathcal{O}^{(l+2)} \quad \vdash \mathbf{L}^{(l)} : \mathcal{L}(\mathcal{O}^{(l)})$$

so that $\Lambda \vdash J^{(l)} \alpha \mathbf{L}^{(l)} : \mathcal{O}^{(l+2)}$ when $\Lambda \vdash \alpha : \mathcal{O}$. In particular, $\Lambda \vdash J \alpha \mathbf{L} : \mathcal{O}''$ for such α .

These constructs give examples of higher level canonical terms.

8.4 LEMMA. *For each $k < \omega$, if $\alpha \in \mathcal{C}_0$ then $J^{(k)} \alpha \mathbf{L}^{(k)} \in \mathcal{C}_{k+2}$.*

Proof. It suffices to show $J^{(k)} \alpha \mathbf{L}^{(k)} \in \mathcal{C}_{k+2}$ for each algebraic α . To this end we fix $C \in \mathcal{C}_{k+1}$, $B \in \mathcal{C}_k$ and show $J^{(k)} \alpha \mathbf{L}^{(k)} C B \in \mathcal{C}_k$ by induction on the structure of α .

Since $J^{(k)} \bar{0} \mathbf{L}^{(k)} C B \triangleright B$ the base case, $\alpha = \bar{0}$, is immediate.

For the induction step $\alpha \mapsto \bar{S}\alpha$ we have

$$J^{(k)}(\bar{S}\alpha) \mathbf{L}^{(k)} C B \triangleright C(J^{(k)} \alpha \mathbf{L}^{(k)} C B)$$

so the required result follows by the induction hypothesis and the given property of C .

For the induction leap $\alpha = \text{Lim} p$ we show

$$J^{(k)} \alpha \mathbf{L}^{(k)} C B A_k \cdots A_1 \zeta \in \mathcal{C}_0$$

where $A_i \in \mathcal{C}_i$ for $k > i \geq 1$ and $\zeta \in \mathcal{C}_0$. Let

$$q = \lambda u : \mathcal{N} . J^{(k)}(pu) \mathbf{L}^{(k)} C B A_k \cdots A_1 \zeta$$

so that $\vdash q : \mathcal{N} \rightarrow \mathcal{O}$. For each $m \in \mathbb{N}$ the term $p\bar{m}$ is canonical so that

$$q\bar{m} \triangleright J^{(k)}(p\bar{m}) \mathbf{L}^{(k)} C B A_k \cdots A_1 \zeta \in \mathcal{C}_0$$

by the induction hypothesis. Also

$$J^{(k)} \alpha \mathbf{L}^{(k)} C B A_k \cdots A_1 \zeta \triangleright \mathbf{L}^{(k)}(\lambda u : \mathcal{N} . J^{(k)}(pu) \mathbf{L}^{(k)} C B) A_k \cdots A_1 \zeta \triangleright \text{Lim} q$$

which leads to the required result. ■

These lifted iterators will enable us to produce many canonical terms. In particular, we will produce canonical names for each ordinal below Howard's ordinal. However, it will take a bit of time before we get up there.

9 The arithmetic of iterators in $\lambda\mathbf{H}$

Suppose we have

$$\Lambda \vdash l : \mathcal{L}(\sigma)$$

for some type σ , context Λ , and term l . We think of this as a local environment in which we can name some iteration gadgets. Of course, we may use the lifts l', l'', \dots to do this.

We say a term α is available in Λ if

$$\Lambda \vdash \alpha : \mathcal{O}$$

holds. For each such α let

$$\mathcal{J}(\alpha) \text{ abbreviate } (\mathbf{J}_\sigma \alpha l)$$

so that

$$\Lambda \vdash \mathcal{J}(\alpha) : \sigma''$$

and, so we claim, the term $\mathcal{J}(\alpha)$ names an iteration gadget within this local environment. We use this abbreviation ‘ $\mathcal{J}(\alpha)$ ’ to hide the details of the environment (which we think of as fixed). In the same way we let

$$\mathcal{J}'(\alpha) \text{ abbreviate } (\mathbf{J}_{\sigma'} \alpha l') \quad \mathcal{J}''(\alpha) \text{ abbreviate } (\mathbf{J}_{\sigma''} \alpha l'') \quad \dots$$

with higher level analogues if we need them. This attaches to α a whole chain of iteration gadgets on various levels above σ .

More generally, for any pair of terms α, β available in Λ , we may form

$$\mathcal{J}(\alpha), \mathcal{J}'(\alpha), \mathcal{J}''(\alpha), \dots, \mathcal{J}(\beta), \mathcal{J}'(\beta), \mathcal{J}''(\beta), \dots$$

and these may be combined in several ways to form more complicated terms. Let

$$\begin{array}{ll} \mathcal{A}(\beta, \alpha) & \text{abbreviate } \lambda y : \sigma', x : \sigma. (\mathcal{J}(\alpha)y)(\mathcal{J}(\beta)yx) \\ \mathcal{M}(\beta, \alpha) & \text{abbreviate } \lambda y : \sigma'. \mathcal{J}(\alpha)(\mathcal{J}(\beta)y) \\ \mathcal{E}(\beta, \alpha) & \text{abbreviate } \mathcal{J}'(\alpha)\mathcal{J}(\beta) \end{array}$$

to obtain three terms $\mathcal{I}(\beta, \alpha)$ such that

$$\Lambda \vdash \mathcal{I}(\beta, \alpha) : \sigma''$$

holds. A few informal calculations with particular cases should convince you that $\mathcal{I}(\beta, \alpha)$ names some kind of iteration gadget on σ .

Now, if as we claim, the system $\lambda\mathbf{H}$ can produce all iteration gadgets (below a certain complexity) as inhabitants of \mathcal{O} , then we should be able to convert the pair α, β into a term γ where

$$\Lambda \vdash \gamma : \mathcal{O} \quad \Lambda \vdash [\mathcal{I}(\beta, \alpha) \approx \mathcal{J}(\gamma)] : \sigma''$$

hold. In this section we show how to produce γ , at least for the more amenable α and β . You can probably guess the results. Informally we may write

$$\begin{array}{lll} \text{(a)} & \mathcal{A}(\beta, \alpha)yx \approx \mathcal{J}(\alpha)y(\mathcal{J}(\beta)yx) \approx \mathcal{J}(\beta + \alpha)yx \\ \text{(m)} & \mathcal{M}(\beta, \alpha)y \approx \mathcal{J}(\alpha)(\mathcal{J}(\beta)y) \approx \mathcal{J}(\beta \cdot \alpha)y \\ \text{(e)} & \mathcal{E}(\beta, \alpha) \approx \mathcal{J}'(\alpha)\mathcal{J}(\beta) \approx \mathcal{J}(\beta^\alpha) \end{array}$$

for certain terms α and β . Here we are using the formal equality symbol \approx in an informal way. We will do this several times in this and the next section. Also

$$'\beta + \alpha' \quad '\beta \cdot \alpha' \quad '\beta^\alpha'$$

are abbreviations which suggest what is going on. We will make these informal claims precise in Theorem 9.5. In particular, we show that if we interpret α and β as ordinals then we retrieve the standard arithmetic with some of the usual rules. However, as in section 6, the arithmetic of iteration templates has fewer identities, so we must proceed with some caution.

In the next section we show how the informal notations of section 6 can be immersed in $\lambda\mathbf{H}$. This will not involve a polymorphic simulation as it did for $\lambda\mathbf{G}$. To do that we need to delve into some of the simpler mechanisms of $\lambda\mathbf{H}$, and that is the topic of this section. This section contains several examples of the equational reasoning facilities of $\lambda\mathbf{H}$, as described in section 8.

We begin with the analogues of addition, multiplication, and exponentiation.

9.1 DEFINITION. Let

$$\begin{aligned} \bar{A} &= \lambda\beta, \alpha : \mathcal{O}. (\text{J}_{\mathcal{O}}\alpha\text{Lim})\bar{S}\beta \\ \bar{M} &= \lambda\beta, \alpha : \mathcal{O}. (\text{J}_{\mathcal{O}}\alpha\text{Lim})(\widehat{A}\beta)\bar{0} \quad \text{where } \widehat{A}\beta = \lambda\gamma : \mathcal{O}. \bar{A}\gamma\beta \\ \bar{E} &= \lambda\beta, \alpha : \mathcal{O}. (\text{J}_{\mathcal{O}}\alpha\text{Lim})(\widehat{M}\beta)\bar{1} \quad \text{where } \widehat{M}\beta = \lambda\gamma : \mathcal{O}. \bar{M}\gamma\beta \end{aligned}$$

to produce three terms $\vdash \bar{O} : \mathcal{O} \rightarrow \mathcal{O} \rightarrow \mathcal{O}$. Here, of course, $\bar{1}$ is $\bar{S}\bar{0}$. ■

These terms should be compared with the numeric terms $\underline{A}, \underline{M}, \underline{E}$ of section 7. In particular, both \bar{M} and \bar{E} include a twist component to swap round the position of two arguments. Here this is more important for even ordinal addition (it is not commutative, and certainly the composition of two iterations is order sensitive. (The reason for letting $\bar{A}, \bar{M}, \bar{E}$ consume their arguments in this order is to simplify certain constructions later.)

We need some justification for calling these terms addition, multiplication, and exponentiation. In the final analysis, this justification should be that they name the corresponding operations on \mathbb{O} . Unfortunately, we don't yet know enough about \mathbb{O} , so we have to proceed differently. In fact, what we are doing here is determining what these operations on \mathbb{O} should be.

Recall that (as defined in section 8) we have the relation \bowtie of reduction equivalence.

9.2 LEMMA. *Suppose*

$$\Lambda \vdash \alpha : \mathcal{O} \quad \Lambda \vdash \beta : \mathcal{O} \quad \Lambda \vdash p : (\mathcal{N} \rightarrow \mathcal{O})$$

are derivable for some context Λ and terms α, β, p . Then each of the following reductions

(Base)	(Step)	(Leap)
$\bar{A}\beta\bar{0} \triangleright \beta$	$\bar{A}\beta(\bar{S}\alpha) \bowtie \bar{S}(\bar{A}\beta\alpha)$	$\bar{A}\beta(\text{Limp}) \bowtie \text{Lim}(\lambda u : \mathcal{N}. (\bar{A}\beta(pu)))$
$\bar{M}\beta\bar{0} \triangleright \bar{0}$	$\bar{M}\beta(\bar{S}\alpha) \bowtie \bar{A}(\bar{M}\beta\alpha)\beta$	$\bar{M}\beta(\text{Limp}) \bowtie \text{Lim}(\lambda u : \mathcal{N}. (\bar{M}\beta(pu)))$
$\bar{E}\beta\bar{0} \triangleright \bar{1}$	$\bar{E}\beta(\bar{S}\alpha) \bowtie \bar{M}(\bar{E}\beta\alpha)\beta$	$\bar{E}\beta(\text{Limp}) \bowtie \text{Lim}(\lambda u : \mathcal{N}. (\bar{E}\beta(pu)))$

hold.

Proof. These are three instances of a whole family of similar results. Let's look at a general case. This will cover the second and third lines (concerning \bar{M} and \bar{E}) and a simple modification covers the first line.

Suppose we have some term

$$\vdash F : (\mathcal{O} \rightarrow \mathcal{O} \rightarrow \mathcal{O})$$

and let

$$G ::= \lambda \delta, \beta, \alpha : \mathcal{O} . (J_{\mathcal{O}} \alpha \text{Lim})(\widehat{F}\beta)\delta \quad \text{where} \quad \widehat{F}\beta = \lambda \gamma : \mathcal{O} . F\gamma\beta$$

to produce a new term

$$\vdash G : (\mathcal{O} \rightarrow \mathcal{O} \rightarrow \mathcal{O} \rightarrow \mathcal{O})$$

with one extra argument. Usually we fix the argument δ as a constant (such as $\bar{0}$ or $\bar{1}$). We show that

$$G\delta\beta\bar{0} \triangleright \delta \quad G\delta\beta(\bar{S}\alpha) \bowtie F(G\delta\beta\alpha)\beta \quad G\delta\beta(\text{Lim}p) \bowtie \text{Lim}(\lambda u : \mathcal{N} . G\delta\beta(pu))$$

hold (for all α, β, δ and p).

(Base) Using a reduction properties of $J_{\mathcal{O}}$ we have

$$G\delta\beta\bar{0} \triangleright (J_{\mathcal{O}}\bar{0}\text{Lim})(\widehat{F}\beta)\delta \triangleright \delta$$

as required.

(Step) In the same way we have

$$G\delta\beta\alpha \triangleright (J_{\mathcal{O}}\alpha\text{Lim})\widehat{F}\beta\delta$$

so that

$$G\delta\beta(\bar{S}\alpha) \triangleright (J_{\mathcal{O}}(\bar{S}\alpha)\text{Lim})(\widehat{F}\beta)\delta \triangleright (\widehat{F}\beta)((J_{\mathcal{O}}\alpha\text{Lim})(\widehat{F}\beta)\delta) \triangleright F((J_{\mathcal{O}}\alpha\text{Lim})(\widehat{F}\beta)\delta)\beta$$

to give the required result.

(Leap) We have

$$G\delta\beta(\text{Lim}p) \triangleright (J_{\mathcal{O}}(\text{Lim}p)\text{Lim})(\widehat{F}\beta)\delta \triangleright \text{Lim}(\lambda u : \mathcal{N} . (J_{\mathcal{O}}(pu)\text{Lim})(\widehat{F}\beta)\delta)$$

which leads to the required result. ■

These results are enough to show that the ordinal operations

$$\mathfrak{Ord}[\bar{A}] \quad \mathfrak{Ord}[\bar{M}] \quad \mathfrak{Ord}[\bar{E}]$$

are just ordinal addition, multiplication, and exponentiation, respectively. Our job here is to explain what $\bar{A}, \bar{M}, \bar{E}$ do on \mathfrak{O} .

We will analyse how these operations behave on those members of \mathfrak{O} that can be named by algebraic terms in the sense of Definition 8.2. It is not clear to me whether this covers the whole of \mathfrak{O} or merely a part. However, it is the best that we can do at the moment.

9.3 THEOREM. For each pair α, β of algebraic terms, there are algebraic terms

$$A(\beta, \alpha) \quad M(\beta, \alpha) \quad E(\beta, \alpha)$$

such that the reductions

$$\bar{A}\beta\alpha \bowtie A(\beta, \alpha) \quad \bar{M}\beta\alpha \bowtie M(\beta, \alpha) \quad \bar{E}\beta\alpha \bowtie E(\beta, \alpha)$$

hold.

Proof. We produce the algebraic terms by recursion on the (syntactic) structure of α with β held rigid. Thus we set

Base	Step	Leap
$A(\beta, \bar{0}) = \beta$	$A(\beta, \bar{S}\alpha) = \bar{S}A(\beta, \alpha)$	$A(\beta, \text{Limp}) = \text{Lim}(\lambda u : \mathcal{N} . \bar{A}\beta(pu))$
$M(\beta, \bar{0}) = \bar{0}$	$M(\beta, \bar{S}\alpha) = A(M(\beta, \alpha), \beta)$	$M(\beta, \text{Limp}) = \text{Lim}(\lambda u : \mathcal{N} . \bar{M}\beta(pu))$
$E(\beta, \bar{0}) = \bar{1}$	$E(\beta, \bar{S}\alpha) = M(E(\beta, \alpha), \beta)$	$E(\beta, \text{Limp}) = \text{Lim}(\lambda u : \mathcal{N} . \bar{E}\beta(pu))$

for appropriate α, β , and p . We must show that each such term is algebraic and exhibit appropriate reductions. Of course, we proceed by induction on the structure of α . (We will see what this means in a moment.)

We look at the three cases (a, m, e) in turn.

(a) Since β is algebraic and $\bar{A}\beta\bar{0} \triangleright \beta$, the base case, $\alpha = \bar{0}$, is immediate.

For the step case, $\alpha \mapsto \bar{S}\alpha$, the induction hypothesis ensures that $A(\beta, \bar{S}\alpha) = \bar{S}A(\beta, \alpha)$ is algebraic, and

$$\bar{A}\beta(\bar{S}\alpha) \bowtie \bar{S}(\bar{A}\beta\alpha) \bowtie \bar{S}A(\beta, \alpha) = A(\beta, \bar{S}\alpha)$$

by Lemma 9.2 and a second use of the induction hypothesis.

The induction leap to an algebraic term $\alpha = \text{Limp}$ is a little more interesting. Let

$$q = \lambda u : \mathcal{N} . \bar{A}\beta(pu)$$

so that $A(\beta, \text{Limp}) = \text{Lim}q$ is the constructed term. In particular, Lemma 9.2 gives

$$\bar{A}\beta(\text{Limp}) \bowtie \text{Lim}q = A(\beta, \text{Limp})$$

so it suffices to show that $\text{Lim}q$ is algebraic.

By assumption, the term Limp is algebraic. Thus, that for each $m \in \mathbb{N}$ there is some algebraic term α_m such that

$$\vdash [p\underline{m} \approx \alpha_m] : \mathcal{O} \quad \bar{A}\beta\alpha_m \bowtie A(\beta, \alpha_m) \quad A(\beta, \alpha_m) \text{ is algebraic}$$

hold. The second and third of these follow by the induction hypothesis applied to α_m . These give

$$\vdash [\bar{A}\beta(p\underline{m}) \approx \bar{A}\beta\alpha_m] : \mathcal{O} \quad \vdash [\bar{A}\beta\alpha_m \approx A(\beta, \alpha_m)] : \mathcal{O}$$

and hence

$$\vdash [q\underline{m} \approx A(\beta, \alpha_m)] : \mathcal{O}$$

(for each $m \in \mathbb{N}$) which is enough to show that $\text{Lim}q$ is algebraic, as required.

[Note that we do *not* set

$$A(\beta, \text{Limp}) = \text{Lim}(\lambda u : \mathcal{N} . A(\beta, pu))$$

since this does not make sense. Firstly, the term ‘ pu ’ is not algebraic so ‘ $A(\beta, pu)$ ’ is not defined. Secondly, the terms $A(\beta, \alpha)$ are constructed by an external recursion on the syntax of α , and we can not internalize this construction to get ‘ $A(\beta, pu)$ ’ as a single term applied to u .]

(m) The base case, $\alpha = \bar{0}$, is immediate.

For the step case, $\alpha \mapsto \bar{S}\alpha$, we have

$$\bar{M}\beta(\bar{S}\alpha) \bowtie \bar{A}(\bar{M}\beta\alpha)\beta \bowtie \bar{A}M(\beta, \alpha)\beta \bowtie A(M(\beta, \alpha), \beta) = M(\beta, \bar{S}\alpha)$$

as required. The first equivalence follows by Lemma 9.2, the second by the induction hypothesis (since $M(\beta, \alpha)$ is algebraic), and the third follows by part (a).

For the induction leap to an algebraic term $\alpha = \text{Limp}$ let

$$q = \lambda u : \mathcal{N} . \bar{M}\beta(pu)$$

so that $M(\beta, \text{Limp}) = \text{Lim}q$, by definition, and hence

$$\bar{M}\beta(\text{Limp}) \bowtie M(\beta, \text{Limp})$$

by Lemma 9.2. Thus it suffices to show that $\text{Lim}q$ is algebraic.

By assumption, the term Limp is algebraic. Thus for $m \in \mathbb{N}$ there is some algebraic term α_m such that

$$pm \approx \alpha_m \quad \bar{M}\beta\alpha_m \bowtie M(\beta, \alpha_m) \quad M(\beta, \alpha_m) \text{ is algebraic}$$

where we have use a slight abbreviations in the left hand clause. These give

$$qm \bowtie \bar{M}\beta(pm) \approx \bar{M}\beta\alpha_m \approx M(\beta, \alpha_m)$$

and hence

$$\vdash [qm \approx M(\beta, \alpha_m)] : \mathcal{O}$$

(for each $m \in \mathbb{N}$) which is enough to show that $\text{Lim}q$ is algebraic, as required.

(e) The proofs for this case are similar, and can be left as an exercise. ■

These equivalences are stated in terms of algebraic terms, but the same results hold for canonical terms (since, up to \approx , each canonical term can be replaced by an algebraic term.) As a consequence of this we have further examples of the class \mathcal{C}_1 of terms of Definition 8.3.

9.4 COROLLARY. *For each canonical term β the three terms $\bar{A}\beta, \bar{M}\beta, \bar{E}\beta$ belong to \mathcal{C}_1 .*

We have the problem of converting terms α, β into a term γ such that $\mathcal{I}(\beta, \alpha) \approx \mathcal{J}(\gamma)$ holds. We can now do that, at least for the case where α and β are algebraic. This result makes precise the three assertions (a, m, e) at the beginning of this section.

9.5 THEOREM. Let Λ be a context with

$$\Lambda \vdash l : \mathcal{L}(\sigma)$$

for some type σ and term l . Then

- (a) $\Lambda \vdash [\mathcal{A}(\beta, \alpha) \approx \mathcal{J}(\bar{\mathbf{A}}\beta\alpha)] : \sigma''$
- (m) $\Lambda \vdash [\mathcal{M}(\beta, \alpha) \approx \mathcal{J}(\bar{\mathbf{M}}\beta\alpha)] : \sigma''$
- (e) $\Lambda \vdash [\mathcal{E}(\beta, \alpha) \approx \mathcal{J}(\bar{\mathbf{E}}\beta\alpha)] : \sigma''$

hold for all algebraic terms α, β .

Proof. Each of these is proved by induction over the structure of α with β as a rigid parameter. Most of the calculations are done in the context

$$\Gamma ::= \Lambda, y : \sigma', x : \sigma$$

where x, y are fresh identifiers. Let's look at a selection of the details. In these calculations we use the reduction equivalences

$$\begin{array}{lll} \text{(Obs)} & \mathcal{J}(\bar{0})sr \bowtie r & \mathcal{J}(\bar{\mathbf{S}}\alpha)sr \bowtie s(\mathcal{J}(\alpha)sr) & \mathcal{J}(\text{Limp})sr \bowtie l(\lambda u : \mathcal{N}. \mathcal{J}(pu)sr) \\ & \mathcal{J}'(\bar{0})ts \bowtie s & \mathcal{J}'(\bar{\mathbf{S}}\alpha)ts \bowtie t(\mathcal{J}'(\alpha)ts) & \mathcal{J}'(\text{Limp})ts \bowtie l'(\lambda u : \mathcal{N}. \mathcal{J}'(pu)ts) \end{array}$$

each of which can be verified by unravelling the definition of \mathcal{J} of \mathcal{J}' .

(a) (Base) For the base case, $\alpha = \bar{0}$, we have

$$\mathcal{A}(\beta, \bar{0})yx \bowtie (\mathcal{J}(\bar{0})y)(\mathcal{J}(\beta)yx) \bowtie \mathcal{J}(\beta)yx \quad \bar{\mathbf{A}}\beta\bar{0} \bowtie \beta$$

to give the required result. Here the first equivalence on the left unravels the definition of $\mathcal{A}(\beta, \bar{0})$, and the second uses one of (Obs) above. The equivalence on the right is from Lemma 9.2.

(Step) For the induction step, $\alpha \mapsto \bar{\mathbf{S}}\alpha$, we have

$$\begin{aligned} \mathcal{A}(\beta, \bar{\mathbf{S}}\alpha)yx &\bowtie (\mathcal{J}(\bar{\mathbf{S}}\alpha)y)(\mathcal{J}(\beta)yx) \\ &\bowtie y((\mathcal{J}(\alpha)y)(\mathcal{J}(\beta)yx)) \bowtie y(\mathcal{A}(\beta, \alpha)yx) \\ \mathcal{J}(\bar{\mathbf{A}}\beta(\bar{\mathbf{S}}\alpha))yx &\bowtie \mathcal{J}(\bar{\mathbf{S}}(\bar{\mathbf{A}}\beta\alpha))yx \quad \bowtie y(\mathcal{J}(\bar{\mathbf{A}}\beta\alpha)yx) \end{aligned}$$

so that a use of the induction hypothesis gives the required result. In the top block of equivalences, the first follows by unravelling the definition of \mathcal{A} , the second by one of (Obs) above, and the third is another unravelling of \mathcal{A} . In the top block of equivalences, the first follows by Lemma 9.2, and the second is another use of (Obs).

(Leap) For the induction leap to Limp where $\vdash p : \mathcal{N} \rightarrow \mathcal{O}$ let

$$\phi = \lambda u : \mathcal{N}. \mathcal{A}(\beta, pu)yx \quad \psi = \lambda u : \mathcal{N}. \mathcal{J}(\mathbf{A}\beta(pu))yx$$

so that

$$\Gamma \vdash \phi : (\mathcal{N} \rightarrow \mathcal{O}) \quad \Gamma \vdash \psi : (\mathcal{N} \rightarrow \mathcal{O})$$

where Γ is the context given above. We have

$$\begin{aligned} \mathcal{A}(\beta, \text{Limp})yx &\bowtie (\mathcal{J}(\text{Limp})y)(\mathcal{J}(\beta)yx) &\bowtie l(\lambda u : \mathcal{N} . (\mathcal{J}(pu)y)(\mathcal{J}(\beta)yx)) &\bowtie l\phi \\ \mathcal{J}(\bar{\mathbf{A}}\beta(\text{Limp}))yx &\bowtie \mathcal{J}(\text{Lim}q)yx &\bowtie l(\lambda u : \mathcal{N} . \mathcal{J}(qu)yx) &\bowtie l\psi \end{aligned}$$

where

$$q = \lambda u : \mathcal{N} . (\bar{\mathbf{A}}\beta(pu))$$

in the lower block. The top block of equivalences follows by unravelling the definition of \mathcal{A} , one of (Obs) above, and another unravelling of \mathcal{A} . The bottom block of equivalences follows by Lemma 9.2, a use of (Obs), and another unravelling of \mathcal{A} . Now, for each $m \in \mathbb{N}$ there is an algebraic term α_m with

$$\vdash [p\underline{m} \approx \alpha_m] : \mathcal{O}$$

and then

$$\Gamma \vdash [\phi\underline{m} \approx \mathcal{A}(\beta, \alpha_m)yx] : \sigma \quad \Gamma \vdash [\psi\underline{m} \approx \mathcal{J}(\mathbf{A}\beta\alpha_m)yx] : \sigma$$

so that

$$\Gamma \vdash [\phi\underline{m} \approx \psi\underline{m}] : \sigma$$

follows by the induction hypothesis. The ω -rule gives

$$\Gamma \vdash [\phi \approx \psi] : (\mathcal{N} \rightarrow \sigma'')$$

so that

$$\Gamma \vdash [l\phi \approx l\psi] : \sigma''$$

which leads to the required result.

Notice that for these arguments we do not need that β is algebraic. However, we will need this in (m,e).

(m) As indicated, we proceed by induction over α , and we follow the general format of the proof of (a).

(Base) This is straight forward.

(Step) For the induction step, $\alpha \mapsto \bar{\mathbf{S}}\alpha$, we have

$$\begin{aligned} \mathcal{M}(\beta, \bar{\mathbf{S}}\alpha)yx &\bowtie \mathcal{J}(\bar{\mathbf{S}}\alpha)(\mathcal{J}(\beta)y)x &\bowtie \mathcal{J}(\beta)y(\mathcal{J}(\alpha)(\mathcal{J}(\beta)y)x) &\bowtie \mathcal{J}(\beta)y(\mathcal{M}(\beta, \alpha)yx) \\ \mathcal{J}(\bar{\mathbf{M}}\beta(\bar{\mathbf{S}}\alpha))yx &\bowtie \mathcal{J}(\bar{\mathbf{A}}((\bar{\mathbf{M}}\beta\alpha)\beta))yx &\approx \mathcal{A}((\bar{\mathbf{M}}\beta\alpha, \beta))yx &\bowtie \mathcal{J}(\beta)y(\mathcal{J}(\bar{\mathbf{M}}\beta\alpha)yx) \end{aligned}$$

so that a use of the induction hypothesis gives the required result. The top block of equivalences follow by unravelling the definition of \mathcal{M} , a use of one of (Obs) above, and another unravelling of \mathcal{M} . The bottom block of equivalences follow by Lemma 9.2, part (a), and unravelling of \mathcal{A} . As we have seen, in the use of part (a) we need to know that β is algebraic, but we don't need to know that of $\bar{\mathbf{M}}\beta\alpha$.

(Leap) This is similar to the corresponding case for (a), and for (e) which we will give in full.

(e) Since this is potentially the most complicated case, let's look at all the details.

(Base) For the base case, $\alpha = \bar{0}$, we have

$$\mathcal{E}(\beta, \bar{0})yx \bowtie \mathcal{J}'(\bar{0})\mathcal{J}(\beta)yx \bowtie yx \quad \bar{\mathbf{E}}\beta\bar{0} \bowtie \bar{1}$$

by an unravelling of \mathcal{E} , one of (Obs), and Lemma 9.2. But

$$\mathcal{J}(\bar{1})yx = \mathcal{J}(\bar{S0})yx \bowtie yx$$

and this leads to the required result.

(Step) For the induction step, $\alpha \mapsto \bar{S}\alpha$, we have

$$\begin{aligned} \mathcal{E}(\beta, \bar{S}\alpha)y \bowtie \mathcal{J}'(\bar{S}\alpha)\mathcal{J}(\beta)y &\bowtie \mathcal{J}(\beta)(\mathcal{J}'(\alpha)\mathcal{J}(\beta)y) \bowtie \mathcal{J}(\beta)(\mathcal{E}(\beta, \alpha)y) \\ \mathcal{J}(\bar{E}\beta(\bar{S}\alpha))y \bowtie \mathcal{J}(\bar{M}((\bar{E}\beta\alpha)\beta))y &\approx \mathcal{M}((\bar{E}\beta\alpha, \beta)y) \bowtie \mathcal{J}(\beta)(\mathcal{J}(\bar{E}\beta\alpha)y) \end{aligned}$$

so that a use of the induction hypothesis gives the required result. The top block of equivalences follow by by unravelling the definition of \mathcal{E} , a use of one of (Obs), and another unravelling of \mathcal{E} . The bottom block of equivalences follow by Lemma 9.2, part (m), and unravelling of \mathcal{M} . For the use of part (m) we need to know that both β and $\bar{E}\beta\alpha$ are algebraic. The first of these is algebraic by hypothesis, but the second is not. However, by Theorem 9.3, the term is canonical and can be replaced by an algebraic equivalent. This is one reason why we proved Theorem 9.3.

(Leap) For the induction leap to Limp where $\vdash p : \mathcal{N} \rightarrow \mathcal{O}$ let

$$\phi = \lambda u : \mathcal{N} . \mathcal{E}(\beta, pu)yx \quad \psi = \lambda u : \mathcal{N} . \mathcal{J}(\bar{E}\beta(pu))yx$$

so that

$$\Gamma \vdash \phi : (\mathcal{N} \rightarrow \mathcal{O}) \quad \Gamma \vdash \psi : (\mathcal{N} \rightarrow \mathcal{O})$$

where Γ is the context given above. We have

$$\begin{aligned} \mathcal{E}(\beta, \text{Limp})yx &\bowtie \mathcal{J}'(\text{Limp})\mathcal{J}(\beta)yx \\ &\bowtie l'(\lambda u : \mathcal{N} . \mathcal{J}'(pu)\mathcal{J}(\beta)y)x \\ &\bowtie l(\lambda u : \mathcal{N} . \mathcal{J}'(pu)\mathcal{J}(\beta)yx) \bowtie l\phi \\ \mathcal{J}(\bar{E}\beta(\text{Limp}))yx &\bowtie \mathcal{J}(\text{Limq})yx \\ &\bowtie l(\lambda u : \mathcal{N} . \mathcal{J}(qu)yx) \bowtie l\psi \end{aligned}$$

where

$$q = \lambda u : \mathcal{N} . (\bar{E}\beta(pu))$$

in the lower block. The top block of equivalences follows by by unravelling the definition of \mathcal{E} , one of (Obs), the definition of the lift l' , and another unravelling of \mathcal{E} . The bottom block of equivalences follows by Lemma 9.2, a use of one of (Obs), and another unravelling of \mathcal{A} . Now, for each $m \in \mathbb{N}$ there is an algebraic term α_m with

$$\vdash [p\underline{m} \approx \alpha_m] : \mathcal{O}$$

and then

$$\Gamma \vdash [\phi\underline{m} \approx \mathcal{E}(\beta, \alpha_m)yx] : \sigma \quad \Gamma \vdash [\psi\underline{m} \approx \mathcal{J}(\bar{E}\beta\alpha_m)yx] : \sigma$$

so that

$$\Gamma \vdash [\phi\underline{m} \approx \psi\underline{m}] : \sigma$$

follows by the induction hypothesis. The ω -rule gives

$$\Gamma \vdash [\phi \approx \psi] : (\mathcal{N} \rightarrow \sigma'')$$

so that

$$\Gamma \vdash [l\phi \approx l\psi] : \sigma''$$

which leads to the required result.

This completes the full proof. ■

All this may look like a rather pedantic way of doing something which is well known and easy. The problem is that so far all the illustrations have been based on addition, multiplication, and exponentiation of ordinals. Once we start to use more powerful operations and get beyond ϵ_0 things are not so straight forward. It is then that the benefits of this section begin to show.

10 Naming in $\lambda\mathbf{H}$

At this stage it will be useful if we review the various families of notations we have seen, and remember how these determine iteration gadgets.

In Definition 6.1 we introduced the informal notations. These form a small language \mathbf{IN} which is disjoint from the language of $\lambda\mathbf{H}$. The meaning of these notations is given by Definition 6.3, and depends on a nominated function $\varpi[\cdot]$ (needed to interpret the primitive symbol ω).

In Definition 7.2 and Theorem 7.3 we saw how each informal notation is simulated in $\lambda\mathbf{G}$. Thus, we attach to each $\alpha \in \mathbf{IN}$ a whole family of $\lambda\mathbf{G}$ -terms

$$\vdash \alpha_\sigma : \mathcal{L}(\sigma) \rightarrow \sigma''$$

indexed by the types σ of $\lambda\mathbf{G}$.

On several occasions we have claimed that each $\lambda\mathbf{H}$ -term

$$\vdash \alpha : \mathcal{O}$$

names an iteration gadget, and in section 9 we began to see how this works. We concentrated on the algebraic terms, or the essentially equivalent canonical terms, as given by Definition 8.2. Using Lemma 8.4 and Corollary 9.4 we have various ways of constructing such terms. In this section we take a closer look at these algebraic terms.

We begin by translating each informal notation into an algebraic term. To do this we will use the constructions of Theorem 9.3, but before that we need a bit of preparation.

The meaning of each $\alpha \in \mathbf{IN}$ (as generated by Definition 6.3) depends on a globally chosen function $\varpi[\cdot] : \mathbb{N}'$. We need to reflect this choice in $\lambda\mathbf{H}$ (just as we did with simulation of into $\lambda\mathbf{G}$). Thus we assume given some $\lambda\mathbf{H}$ -term

$$\vdash \varpi : \mathcal{N}'$$

which names $\varpi[\cdot]$. This may be the same term as used in the simulation or it may not, since $\lambda\mathbf{H}$ has many more possible terms.

Given this term ϖ , which term do we use to translate ω ? The obvious choice is $\text{Lim}\varpi$, except that this is not well formed. We have

$$\vdash \text{Lim} : (\mathcal{N} \rightarrow \mathcal{O}) \rightarrow \mathcal{O} \quad \vdash \varpi : \mathcal{N} \rightarrow \mathcal{N}$$

so that

$$' \vdash \text{Lim} \varpi : \mathcal{O} '$$

is not derivable. We need some way of immersing \mathcal{N} into \mathcal{O} .

10.1 DEFINITION. Let

$$\text{fin} = \lambda u : \mathcal{N} . l_{\mathcal{O}} u \bar{S} \bar{0}$$

to obtain a term $\vdash \text{fin} : (\mathcal{N} \rightarrow \mathcal{O})$. ■

Recall that for each $m \in \mathbb{N}$ we set

$$\underline{m} = \underline{S}^m \underline{0} \quad \bar{m} = \bar{S}^m \bar{0}$$

to produce the canonical name of m as an inhabitant of \mathcal{N} and \mathcal{O} , respectively. A simple calculation shows that

$$\text{fin} \underline{m} \triangleright \bar{m}$$

holds for each $m \in \mathbb{N}$. With this we can produce the correct translation of ω .

Let

$$\text{fin} \circ \varpi \quad \text{abbreviate} \quad \lambda u : \mathcal{N} . \text{fin}(\varpi u)$$

(the obvious way to ‘compose’ two terms). Since

$$\vdash \text{fin} \circ \varpi : \mathcal{N} \rightarrow \mathcal{O}$$

we see that $\text{Lim}(\text{fin} \circ \varpi)$ is correctly formed.

10.2 DEFINITION. We assign to each informal notation $\alpha \in \mathbf{IN}$ an algebraic term $\bar{\alpha}$ by recursion on the structure of α .

(Zero)	$\bar{0}$ is the primitive symbol		
(Step)	$\bar{S}\alpha = \bar{S}\bar{\alpha}$		
(Omega)	$\bar{\omega} = \text{Lim}(\text{fin} \circ \varpi)$		
(Sum)	$\overline{\zeta + \eta} = A(\bar{\zeta}, \bar{\eta})$	=	‘ $\bar{\zeta} + \bar{\eta}$ ’
(Exp)	$\bar{\omega}^\alpha = E(\bar{\omega}, \bar{\alpha})$	=	‘ $\bar{\omega}^{\bar{\alpha}}$ ’

The (Omega) clause is explained above. The (Sum) and (Exp) clauses make use of the constructions of Theorem 9.3, and the extreme right hand column explains the intention behind these. ■

By Theorem 9.3 we see that each term $\bar{\alpha}$ is algebraic. Only $\bar{\omega}$ requires a little bit of justification. However, for each $m \in \mathbb{N}$ we have

$$(\text{fin} \circ \varpi) \underline{m} \triangleright \bar{n}$$

where $n = \varpi[m]$, which give the required result.

What do these terms do? As an illustration let’s compare

$$\omega_\sigma = \lambda l y x . l(\lambda u : \mathcal{N} . l_\sigma(\varpi u) y x) \quad \bar{\omega} = \text{Lim}(\text{fin} \circ \varpi)$$

where ω_σ is the term produced by Definition 7.2. We will show that

$$\vdash [\omega_\sigma \approx J_\sigma \bar{\omega}] : \mathcal{O}$$

holds.

Let

$$\phi = \lambda u : \mathcal{N} . l_\sigma(\varpi u)yx \quad \psi = \lambda u : \mathcal{N} . J_\sigma(\text{fin}(\varpi u))lyx$$

so that

$$\Gamma \vdash \phi : \mathcal{N} \rightarrow \sigma'' \quad \Gamma \vdash \psi : \mathcal{N} \rightarrow \sigma''$$

where

$$\Gamma = l : \mathcal{L}(\sigma), y : \sigma'.x : \sigma$$

is the context. We have

$$\omega_\sigma lyx \triangleright l\phi \quad J_\sigma \bar{\omega} lyx \triangleright l\psi$$

which shows us what to do. For $m \in \mathbb{N}$ let $n = \varpi[m]$. Then

$$\phi \underline{m} \triangleright y^n x \quad \psi \underline{m} \triangleright J_\sigma(\text{fin} \underline{m}) lyx \triangleright J_\sigma \bar{n} lyx \triangleright y^n x$$

so that a use of the ω -rule leads to the required result.

This illustration is a crucial component in the proof of the following.

10.3 THEOREM. *For each informal notation $\alpha \in \mathbf{IN}$ and each type σ*

$$\vdash [\alpha_\sigma \approx J_\sigma \bar{\alpha}] : \mathcal{O}$$

holds.

Proof. We proceed by induction over the structure of $\alpha \in \mathbf{IN}$. The (Zero) and (Step) case are easy, and we did the (Omega) case above. Let's look at the details of the (Sum) and (Exp) cases.

(Sum) Using Definition 7.2 and Theorem 9.5(a) with Theorem 9.3 we have

$$(\zeta + \eta)_\sigma lyx \triangleright (\eta_\sigma ly)(\zeta_\sigma lyx) \quad J_\sigma A(\bar{\zeta}, \bar{\eta}) lyx \approx (J_\sigma \bar{\eta} ly)(J_\sigma \bar{\zeta} lyx)$$

so two uses of the induction hypothesis gives the required result.

(Exp) Using Definition 7.2 and Theorem 9.5(a) with Theorem 9.3 we have

$$(\omega^\alpha)_\sigma lyx \triangleright (\alpha_{\sigma'} l')(\omega_\sigma l)yx \quad J_\sigma E(\bar{\omega}, \bar{\alpha}) lyx \approx (J_{\sigma'} \bar{\alpha} l')(J_\sigma \bar{\omega} l)yx$$

so two uses of the induction hypothesis gives the required result. ■

Since $\lambda\mathbf{G}$ is a subsystem of $\lambda\mathbf{H}$, anything that can be done in the smaller system can be done in the larger system. Theorem 10.3 shows that many of these things that can be done in $\lambda\mathbf{H}$ in a more uniform fashion. However, it does not show us that more things can be done. In particular, we haven't yet seen named an ordinal beyond ϵ_0 . In the remainder of this section we will develop a the syntactic technique by which $\lambda\mathbf{H}$ can name many much larger ordinals. In the later sections we will calculate the ordinal values of these names.

Before we begin this development proper, let's see how we can get up to just below ϵ_0 but not to ϵ_0 itself.

10.4 EXAMPLE. In Example 7.1 we saw how each of the functions $\mathbb{G}\epsilon[r]ack[change]S$ (for $r \in \mathbb{N}$) could be named in $\lambda\mathbf{G}$. Embedded in that term was a name for the component $\epsilon[r]$ of the fundamental sequence ϵ_0 . These terms can be generated explicitly by a variant of the method of Definition 10.2. Set

$$\epsilon[0, \sigma, l] = \omega_\sigma l \quad \epsilon[r+1, \sigma, l] = \epsilon[r, \sigma', l'](\omega_\sigma l)$$

for each $r \in \mathbb{N}$, type σ , and term l . Note that this is a construction by recursion over r with variation of the two parameters σ, l . It is easy to check that if

$$\Lambda \vdash l : \mathcal{L}(\sigma)$$

then

$$\Lambda \vdash \epsilon[r, \sigma, l] : \sigma'' \quad \Lambda \vdash [\epsilon[r, \sigma, l] \approx \epsilon[r]_\sigma l] : \sigma''$$

where $\epsilon[r]_\sigma$ is generated by repeated use of the (Exp) clause of Definition 7.2. Next set

$$\mathbf{eps} = \lambda u : \mathcal{N} . l_{\mathcal{O}} u(\overline{\mathbf{E}\omega})\overline{\omega}$$

to produce a term where

$$\vdash \mathbf{eps} : \mathcal{N} \rightarrow \mathcal{O}$$

and

$$\mathbf{eps} \underline{r} \bowtie \overline{\epsilon[r]}$$

holds for each $r \in \mathbb{N}$. An application of Theorem 10.3 now gives

$$\Lambda \vdash [\epsilon[r, \sigma, l] \approx J_\sigma(\mathbf{eps} \underline{r})l] : \sigma''$$

for each $r \in \mathbb{N}$. Of course, this can be verified directly by induction on r . ■

The terms $\epsilon[r, \sigma, l]$ give us an indication of why ϵ_0 can not be named in $\lambda\mathbf{G}$ (at least by the obvious method). To produce the terms we need to use higher and higher types, and in the end we run out of types. However, in $\lambda\mathbf{H}$ the term \mathbf{eps} names the chosen fundamental sequence for ϵ_0 , and so it is reasonable to take

$$\overline{\epsilon_0} = \mathbf{Lim} \mathbf{eps}$$

as the name for ϵ_0 itself. Notice that to construct $\overline{\epsilon_0}$ we need nothing like the full power of $\lambda\mathbf{H}$. All we need are $l_{\mathcal{O}}$, $J_{\mathcal{O}}$, and \mathbf{Lim} .

You should remember there are several choices embedded in this name. There is the choice of the function $\varpi[\cdot]$, the choice of term ϖ which names $\varpi[\cdot]$, and the choice of the way the fundamental sequence $\epsilon[\cdot]$ is generated. Thus, to be pedantic, we should say $\mathbf{Lim} \mathbf{eps}$ is a name for that iteration gadget which is performed by \dots , where we give an appropriate description of this.

We will have

$$\mathfrak{Ord}[\overline{\epsilon_0}] = \epsilon_0$$

provided the base function $\varpi[\cdot]$ increases without bound. (You might like to calculate $\mathfrak{Ord}[\overline{\epsilon_0}]$ for the case where $\varpi[r] = 0$ for each $r \in \mathbb{N}$.)

How do we get above this level? We start to generate the analogues of the critical ordinals. To simplify the account let us now choose $\varpi[\cdot]$ to be the identity function, and let us write ω for $\overline{\omega}$ and ϵ for $\overline{\epsilon}$ with this choice.

10.5 DEFINITION. Let

$$\omega = \text{Lim fin} \quad \epsilon = \text{Lim eps} \quad \text{where eps} = \lambda u : \mathcal{N} . \text{I}_{\mathcal{O}}(\bar{\text{E}}\omega)\omega$$

to produce terms $\vdash \omega : \mathcal{O}$ and $\vdash \epsilon : \mathcal{O}$.

Let

$$\text{Fix} = \lambda f : \mathcal{O}' , \alpha : \mathcal{O} . (\text{J}_{\mathcal{O}}\omega\text{Lim})f(\bar{\text{S}}\alpha) \quad \text{Next} = \text{Fix}(\bar{\text{E}}\omega)$$

to produce terms $\vdash \text{Fix} : \mathcal{O}''$ and $\vdash \text{Next} : \mathcal{O}'$. ■

A simple calculation gives

$$\text{Next } \omega \triangleright (\text{J}_{\mathcal{O}}\omega\text{Lim})(\bar{\text{E}}\omega)(\bar{\text{S}}\omega) \triangleright \text{Lim}(\lambda u : \mathcal{N} . (\text{J}_{\mathcal{O}}(\text{fin } u)\text{Lim})(\bar{\text{E}}\omega)(\bar{\text{S}}\omega))$$

which you should compare with

$$\epsilon = \text{Lim eps} \triangleright \text{Lim}(\lambda u : \mathcal{N} . (\text{I}_{\mathcal{O}}u)(\bar{\text{E}}\omega)\omega)$$

from above. The two terms both name the ordinal ϵ_0 but via different fundamental sequences. and so they name different iteration templates.

This is the point where we start to take off.

In the next definition we use the abbreviations $\text{J}^{(\cdot)}$ and $\text{L}^{(\cdot)}$ introduced towards the end of section 8.

10.6 DEFINITION. For each $l < \omega$ let

$$[l] = \lambda y : \mathcal{O}^{(l+1)} , y_l : \mathcal{O}^{(l)} , \dots , x : \mathcal{O} . \text{Fix } Zx \quad \text{where } Z = \lambda \alpha : \mathcal{O} . (\text{J}^{(l)}\alpha\text{L}^{(l)})yy_k \cdots y_1\bar{0}$$

using the auxiliary term on the right.

Let

$$\Delta[0] = \omega \quad \Delta[1] = \text{Next}\omega \quad \Delta[l+2] = [l] \cdots [0]\text{Next}\omega$$

for each $l < \omega$. ■

It isn't hard to see that

$$\vdash [l] : \mathcal{O}^{(l+2)} \quad \vdash \Delta[r] : \mathcal{O}$$

for each $l, r \in \mathbb{N}$. In particular, we have a family of ordinals

$$\mathfrak{Ord}(\Delta[r])$$

(for $r \in \mathbb{N}$). What are these? Almost trivially

$$\mathfrak{Ord}(\Delta[0]) = \omega \quad \mathfrak{Ord}(\Delta[1]) = \epsilon_0$$

and you might be able to calculate (or guess the value of) $\mathfrak{Ord}(\Delta[2])$ and $\mathfrak{Ord}(\Delta[3])$. We show how to calculate all $\mathfrak{Ord}(\Delta[l])$ in next few sections, where we find that these ordinals form a fundamental sequence for the Howard ordinal.

Observe that $\Delta[r]$ is built using $\text{J}^{(0)}, \text{J}^{(1)}, \dots, \text{J}^{(r)}$ within a restricted part of the type hierarchy on \mathcal{O} (and \mathcal{N}). Thus it should be possible to put an upper bound on the complexity of the iteration template and the ordinal named by $\Delta[r]$.

In Definition 8.3 we introduced the hierarchy \mathcal{C}_\bullet of canonical terms, and Lemma 8.4 and Corollary 9.4 gave us several examples of these. We now have a few more.

10.7 LEMMA. *We have*

$$\omega \in \mathcal{C}_0 \quad \epsilon \in \mathcal{C}_0 \quad \text{Fix} \in \mathcal{C}_2 \quad \text{Next} \in \mathcal{C}_1$$

and

$$[l] \in \mathcal{C}_{l+2}$$

for each $l < \omega$.

Proof. For each $m \in \mathbb{N}$ we have

$$\text{fin} \underline{m} \triangleright \bar{m} = \bar{S}^m \bar{0}$$

which, since $\bar{S} \in \mathcal{C}_1$ and $\bar{0} \in \mathcal{C}_0$, gives $\omega \in \mathcal{C}_0$.

Similarly, for each $m \in \mathbb{N}$ we have

$$\text{eps} \underline{m} \triangleright (\bar{E}\omega)^m \omega$$

which, by Corollary 9.4, gives $\epsilon \in \mathcal{C}_0$.

Since

$$\text{Fix } f \alpha \triangleright (J_O \omega \text{Lim}) f(\bar{S}\alpha)$$

a use of Lemma 9.2 gives $\text{Fix} \in \mathcal{C}_2$.

Since $\bar{E}\omega$ the previous part gives $\text{Next} \in \mathcal{C}_1$.

The last part is just a little more interesting. Since

$$[l] CBA_{l-1} \cdots A_1 A_0 \triangleright (J\omega L) Z A_0 \quad \text{where} \quad Z\alpha \triangleright J^{(l)} \alpha L^{(l)} CBA_{l-1} \cdots A_1 0$$

the result follows by two uses of Lemma 9.2. ■

This more or less completes the analysis of λH carried out here. The remainder of the survey is concerned with locating the ordinals $\mathfrak{Ord}(\Delta[l])$ and describing them in different ways.

11 Fruitful and helpful ordinal functions

In this section we look at the class \mathbb{Fruit} of fruitful functions and the layered class \mathbb{Help} of helpful functions.

The fruitful functions are easy to understand. They are a rather mild generalization of the standard normal functions, and are more convenient to deal with.

The helpful functions are not so easy to understand. Even though the notion works I am not entirely convinced that I have the correct definition, and I am certain that other useful facts are waiting to be discovered.

Most of this section is as it was written some time ago. However, since then [FandH] has appeared and this develops the notion in some detail. There is some overlap between that paper and this section, but there is also some information here that is not in [FandH].

Here beginneth the original version.

In this section we construct various particular functions and classes of functions associated with \mathbb{Ord} , the set of countable ordinals. This section can be read independently of all previous sections in the sense that it does not make use of material developed so

far (except without that background it may not be clear why we do certain things). In particular, we produce functions **Fix**, **Next**, $[l]$ and use these to construct ordinals $\Delta[l]$ (for $l < \omega$). In section 12 we show how these match the syntactic terms **Fix**, **Next**, $[l]$ of $\lambda\mathbf{H}$, but at a first reading of this section you should not worry about what that means or how it is done.

The material of this section is developed in some depth in section 7 of $[\lambda\mathbf{H}]$, and we will not repeat those details here. In particular, we will not repeat proofs that are given there. Apart from that this section is self contained.

Most of the function we use are higher order.

11.1 DEFINITION. The chain $\text{Ord}^{(\cdot)}$ of spaces is generated by

$$\text{Ord}^{(0)} = \text{Ord} \quad \text{Ord}^{(r+1)} = \text{Ord}^{(r)'}$$

for each $r < \omega$ (where \mathbb{S}' is $\mathbb{S} \longrightarrow \mathbb{S}$ for each set \mathbb{S}). ■

Thus $\text{Ord}^{(0)}$ is just the space Ord of ordinals, and $\text{Ord}^{(1)}$ is the space Ord' of ordinal functions, and so on.

The space Ord is linearly ordered and carries an actual supremum operation \bigvee which converts each countable subsets $X \subseteq \text{Ord}$ into its least upper bound $\bigvee X$. We lift this to higher levels.

11.2 DEFINITION. (base) For each non-empty, countable subset $\mathcal{G} \subseteq \text{Ord}'$ the function $\bigvee \mathcal{G} : \text{Ord}'$ is given by

$$(\bigvee \mathcal{G})\zeta = \bigvee \{g\zeta \mid g \in \mathcal{G}\}$$

(for $\zeta \in \text{Ord}$). We call this function $\bigvee \mathcal{G}$ the **pointwise supremum** of \mathcal{G} .

(raise) For each $l < \omega$ and each non-empty, countable subset $\mathcal{G} \subseteq \text{Ord}^{(l+2)}$ the function $\bigvee \mathcal{G} : \text{Ord}^{(l+2)}$ is given by

$$(\bigvee \mathcal{G})g = \bigvee \{Gg \mid G \in \mathcal{G}\}$$

(for $g \in \text{Ord}^{(l+1)}$). We call this function $\bigvee \mathcal{G}$ the **pointwise supremum** of \mathcal{G} . ■

Do not be misled by this construction. To explain what is going on let us temporarily write $\bigvee^{(l)}$ for the gadget constructed on $\text{Ord}^{(l)}$. Thus $\bigvee^{(0)}$ is the actual supremum operation on Ord , and then $\bigvee^{(1)}, \bigvee^{(2)}, \dots, \bigvee^{(l)}, \dots$ are constructed in turn by recursion on l . The notation is not intended to suggest there is a partial ordering on $\text{Ord}^{(l)}$ of which $\bigvee^{(l)}$ is the supremum operation (but there is something of this lurking beneath the surface).

11.3 DEFINITION. (a) For each $l < \omega$ and each $g : \text{Ord}^{(l+1)}$, the ordinal iterates g^\bullet of g are generated by

$$g^0 = id \quad g^{\alpha+1} = g \circ g^\alpha \quad g^\lambda = \bigvee \{g^\alpha \mid \alpha < \lambda\}$$

for each $\alpha \in \text{Ord}$ and limit ordinal $\lambda \in \text{Ord}$. (Here id is the identity function on $\text{Ord}^{(l)}$.)

(b) For each $l < \omega$ a class $\mathbb{S} \subseteq \text{Ord}^{(l+1)}$ is **smooth** if $f \circ g \in \mathbb{S}$ for each $f, g \in \mathbb{S}$, and $\bigvee \mathcal{G} \in \mathbb{S}$ for each non-empty and countable $\mathcal{G} \subseteq \mathbb{S}$. ■

Again, for the moment, treat this as nothing more than a definition. In general the ordinal iterates of g (as defined here) may not behave as you think they should. The notion of a smooth class is a way of calming down some of the wilder behaviour.

11.4 LEMMA. Suppose $\mathbb{S} \subseteq \text{Ord}^{(l+1)}$ is smooth and $g \in \mathbb{S}$. Then $g^\alpha \in \mathbb{S}$ for each non-zero ordinal α .

We will construct several smooth classes, some at high levels. However, for the first examples we stick with Ord' . We are going to use several classes of ordinal functions satisfying various combinations of properties. Here are all the properties we need.

11.5 DEFINITION. A function $g : \text{Ord}'$ is, respectively

- | | | |
|------|-----------------------|---|
| (i) | inflationary | if $\alpha \leq g\alpha$ |
| (si) | strictly inflationary | if $\alpha < g\alpha$ |
| (m) | monotone | if $\beta \leq \alpha \Rightarrow g\beta \leq g\alpha$ |
| (sm) | strictly monotone | if $\beta < \alpha \Rightarrow g\beta < g\alpha$ |
| (b) | big | if $\omega^\alpha \leq g\alpha$ (except possibly for $\alpha = 0$) |
| (sb) | strictly big | if $g\alpha$ is critical |
| (c) | continuous | if $g(\bigvee A) = \bigvee g[A]$ |

for all $\alpha, \beta \in \text{Ord}$, and all non-empty countable subsets $A \subseteq \text{Ord}$.

Let \mathbb{IM} be the class of functions that are both inflationary and monotone. ■

Most of these properties are standard, and the relationships between them are well known. The properties ‘big’ and ‘strictly big’ are used to exclude some silly special cases.

As far as Ord' goes we are concerned almost entirely with subclasses of \mathbb{IM} .

11.6 LEMMA. The class \mathbb{IM} is smooth. Furthermore

$$\beta \leq \alpha \implies g^\beta \leq g^\alpha$$

holds for each $g \in \mathbb{IM}$ and ordinals α, β .

Proof. The first part is easy, and the second part follows by induction on α (making use of the inflationary property of g). ■

As with any class of monotone functions the class \mathbb{IM} can be partially ordered using the pointwise comparison.

$$f \leq g \iff (\forall \alpha \in \text{Ord})[f\alpha \leq g\alpha]$$

With this comparison, for each non-empty, countable subset $\mathcal{G} \subseteq \mathbb{IM}$ the pointwise supremum $\bigvee \mathcal{G}$ is the actual supremum, and we find that the following required property of the composites of iterates holds.

11.7 LEMMA. If $g \in \mathbb{IM}$ then

$$g^\alpha \circ g^\beta = g^{\beta+\alpha} \quad (g^\beta)^\alpha = g^{\beta \times \alpha}$$

for all $\alpha, \beta \in \text{Ord}$.

Proof. Both of these are proved by induction on α . Let's look at the leap to a limit ordinal λ for the second identity. Thus we require

$$(g^\beta)^\lambda \zeta = g^{\beta \times \lambda} \zeta$$

for each $\zeta \in \text{Ord}$. We have

$$(g^\beta)^\lambda \zeta = \bigvee \{(g^\beta)^\alpha \zeta \mid \alpha < \lambda\} = \bigvee \{g^{\beta \times \alpha} \zeta \mid \alpha < \lambda\} \quad g^{\beta \times \lambda} \zeta = \bigvee \{g^\gamma \zeta \mid \gamma < \beta \times \lambda\}$$

where the second equality uses the induction hypothesis. Also

$$\beta \times \lambda = \bigvee \{\beta \times \alpha \mid \alpha < \lambda\}$$

(by construction of ordinal multiplication). The comparison

$$(g^\beta)^\lambda \zeta \leq g^{\beta \times \lambda} \zeta$$

is immediate (since $\beta \times \alpha \leq \beta$ for $\alpha < \lambda$). For the converse consider any $\gamma < \beta \times \lambda$. There is some $\alpha < \lambda$ with $\gamma \leq \beta \times \alpha$, and then Lemma 11.6 gives

$$g^\gamma \zeta \leq g^{\beta \times \alpha} \zeta \leq (g^\beta)^\lambda \zeta$$

which leads to the required result. ■

We have made a bit of a meal of this proof to highlight the required properties of g .

As a consequence of this result some limit iterates of functions in \mathbb{IM} are constant for long periods, and so can not be strictly monotone.

11.8 EXAMPLE. Let $g \in \mathbb{IM}$ and suppose λ is additively critical. For each ordinal ζ and ordinal $\alpha < \lambda$ we have

$$g^\lambda(g^\alpha \zeta) = (g^\lambda \circ g^\alpha) \zeta = g^{\alpha + \lambda} \zeta = g^\lambda \zeta$$

and hence g^λ is constant between ζ and $g^\alpha \zeta$.

The situation is even more dramatic if g is continuous. In this case $g^\lambda = g^\mu$ for some $\mu \geq \lambda \cdot \omega$. ■

We use two particular subclasses of \mathbb{IM}

11.9 DEFINITION. An ordinal function $f \in \text{Ord}'$ is, **fruitful** if it is inflationary, monotone, big, and continuous. Let \mathbb{Fruit} be the class of fruitful functions.

An ordinal function $h \in \text{Ord}'$ is, **helpful** if it is strictly inflationary, monotone, and strictly big. Let \mathbb{Help} be the class of helpful functions. ■

There are three main classes of ordinal functions that we use: general functions, fruitful functions, and helpful functions. Usually we write

$$f \text{ for a fruitful function} \quad g \text{ for a general function} \quad h \text{ for a helpful function}$$

to indicate which kind of function is being used. (This convention came about because for a long time I couldn't remember the difference between fruitful and helpful – which were then called something else. I had to invent this little trick to keep my sanity. As it turns out fruitful functions do give us lots of fixed points, and helpful functions make it easy to construct fruitful functions.)

Recall that a function $f \in \text{Ord}'$ is normal if it strictly monotone, big, and continuous. Every such function is fruitful but, as we will see, the larger class \mathbb{Fruit} has much nicer properties.

11.10 LEMMA. *Both the classes \mathbb{Fruit} and \mathbb{Help} are smooth.*

What is the connection between \mathbb{Fruit} and \mathbb{Help} ? To answer that we introduce a particular second level function.

11.11 DEFINITION. Let $\mathbf{Fix} : \text{Ord}''$ be the function given by

$$\mathbf{Fix} f \zeta = f^\omega(\zeta + 1)$$

for each function $f : \text{Ord}'$ and ordinal ζ . ■

We have

$$\mathbf{Fix} f \zeta = \bigvee \{f^r(\zeta + 1) \mid r < \omega\}$$

and this makes sense for any function $f : \text{Ord}'$. However, we use \mathbf{Fix} only on $f \in \mathbb{Fruit}$. For such f we see that \mathbf{Fix} is a fixed point extractor.

11.12 LEMMA. *For each $f \in \mathbb{Fruit}$ and $\zeta \in \text{Ord}$, the value $\mathbf{Fix} f \zeta$ is the least ordinal ν such that $\zeta < \nu = f\nu$.*

Much of the standard material on ordinal notations is about extracting fixed points, so we can see why \mathbf{Fix} might be useful.

11.13 LEMMA. (a) *For each $f \in \mathbb{Fruit}$ the function $\mathbf{Fix} f$ is helpful.*

(b) *For each $h \in \mathbb{Help}$ and ordinal ζ , the ordinal function $\alpha \mapsto h^\alpha \zeta$ is normal.*

According to Definition 11.11 the operator \mathbf{Fix} may act on any ordinal function. However we use it only on fruitful functions so part (a) of this result gives

$$\mathbf{Fix} : \mathbb{Fruit} \longrightarrow \mathbb{Help}$$

or, strictly speaking, the restriction $\mathbf{Fix}|_{\mathbb{Fruit}}$ has this type. This slight confusion will not be a problem.

We need a couple of properties of this function \mathbf{Fix} (in its restricted form). Since $\mathbb{Fruit} \subseteq \mathbb{IM}$ this family carries the pointwise comparison and the pointwise supremum is the actual supremum. Recall also that a family $\mathcal{F} \subseteq \mathbb{Fruit}$ is directed if it is non-empty and for each $f, g \in \mathcal{F}$ there is some $h \in \mathcal{F}$ with $f, g \leq h$. (Here of course, f, g, h are fruitful and h need not be helpful.)

11.14 LEMMA. *The function $\mathbf{Fix} : \mathbb{Fruit} \longrightarrow \mathbb{Help}$ is monotone and Scott-continuous. In other words*

$$f \leq g \Rightarrow \mathbf{Fix} f \leq \mathbf{Fix} g \quad \mathbf{Fix}(\bigvee \mathcal{F}) = \bigvee \{\mathbf{Fix} f \mid f \in \mathcal{F}\}$$

hold for all $f, g \in \mathbb{Fruit}$ and all directed families $\mathcal{F} \subseteq \mathbb{Fruit}$.

A proof of this is not very difficult, but perhaps the need for directedness is not immediately obvious. An example is in order.

11.15 EXAMPLE. We produce a pair f, g of fruitful functions such that

$$\mathbf{Fix}(f \vee g)0 = (\mathbf{Fix} f \vee \mathbf{Fix} g)0$$

fails. Consider

$$\nu < \mu < \epsilon_\nu$$

where both μ, ν are critical. With these set

$$f\alpha = \begin{cases} \epsilon_\alpha & \text{if } \mu < \alpha \\ \mu & \text{if } \alpha \leq \mu \end{cases} \quad g\alpha = \begin{cases} \epsilon_\alpha & \text{if } \nu < \alpha \\ \nu & \text{if } \alpha \leq \nu \end{cases}$$

for each ordinal α to obtain a pair of fruitful functions. We have

$$\mathbf{Fix} f 0 = \mu \quad \mathbf{Fix} g 0 = \nu \quad \mathbf{Fix} f 0 \vee \mathbf{Fix} g 0 = \mu$$

(since $\nu < \mu$). Let $h = f \vee g$. Then

$$h\alpha = \begin{cases} \epsilon_\alpha & \text{if } \mu < \alpha \\ \epsilon_\alpha & \text{if } \nu < \alpha \leq \mu \\ \mu & \text{if } \alpha \leq \nu \end{cases}$$

where only the central clause needs thinking about. But for $\nu < \alpha \leq \mu$ we have

$$h\alpha = \max\{\mu, \epsilon_\alpha\} = \epsilon_\alpha$$

since $\mu < \epsilon_\nu \leq \epsilon_\alpha$. The least fixed point $\lambda = \mathbf{Fix} h 0$ satisfies

$$\nu < \lambda = \epsilon_\lambda$$

and hence

$$\mu < \epsilon_\nu \leq \epsilon_\lambda = \lambda$$

to give the required result. ■

Part (b) of Lemma 11.13 enables us to generate many normal (not just fruitful) functions. By combining both parts we should be able to produce many examples of each kind of function. Of course, we must start somewhere.

11.16 DEFINITION. Let $\mathbf{Next} = \mathbf{Fix} \omega^\bullet$. ■

Since the function ω^\bullet , exponentiation to base ω , is fruitful (in fact, normal) we have the following.

11.17 LEMMA. $\mathbf{Next} \in \mathbb{H}\text{elp}$.

Recall that an ordinal ϵ is critical if $\omega^\epsilon = \epsilon$, a fixed point of ω^\bullet . A simple argument shows that $\mathbf{Next}\zeta$ is the next critical ordinal strictly beyond ζ . In particular

$$\epsilon_\alpha = \mathbf{Next}^\alpha \epsilon_0 = \mathbf{Next}^{1+\alpha} \omega$$

is the long list of critical ordinals. Lemmas 11.17 and 11.13(b) show that this generating function is normal.

It is easy to check that \mathbf{Next} is the smallest helpful function, that is $\mathbf{Next}\zeta \leq h\zeta$ for each $h \in \mathbb{H}\text{elp}$ and ordinal ζ .

Helpful functions have various interesting properties, but here we need only a few of them. The following is Lemma 2.13 of [Sch] where a full proof is given.

11.18 LEMMA. Suppose $h \in \mathbb{H}\text{elp}$. Then

$$(a) \zeta + \alpha \leq h^\alpha \zeta.$$

$$(b) h^\lambda \zeta = h^\lambda 0$$

$$(c) (\zeta < \nu = h^\nu 0) \iff (0 < \nu = h^\nu \zeta)$$

hold for all ordinals α, ν, ζ and (additively) critical ordinal λ with $\zeta < \lambda$.

A standard development of ordinal notation would make much use of normal functions. We will see that fruitful functions are a more amenable way of doing this. The helpful functions provide a canonical way of generating such functions. However, a more important benefit of the notion of helpfulness is that it lifts to higher levels.

Before we see how that is done let's sort out some notational conventions.

By decomposing the space

$$\text{Ord}^{(l+2)} = \text{Ord}^{(l+1)} \rightarrow \text{Ord}^{(l)} \rightarrow \dots \rightarrow \text{Ord}' \rightarrow \text{Ord} \rightarrow \text{Ord}$$

we see that each function $H : \text{Ord}^{(l+2)}$ must receive successive arguments

$$h : \text{Ord}^{(l+1)}, h_l : \text{Ord}^{(l)}, \dots, h_1 : \text{Ord}', \zeta : \text{Ord}$$

to return its eventual ordinal value $Hhh_l \cdots h_1 \zeta$. Often these central arguments h_l, \dots, h_1 play a passive role, so we write $Hhh\zeta$ for the eventual value. We do not use this abbreviation in the following definition, but we will in the subsequent analysis.

11.19 DEFINITION. (Base) Let $\mathbb{H}^{(1)} = \mathbb{H}\text{elp}$, the class of helpful functions on level 1.

(Step) For each $l < \omega$ a function $H : \text{Ord}^{(l+2)}$ is helpful on level $l + 2$ if

$$\text{(Help1)} \quad Hh \text{ is helpful}$$

$$\text{(Help2)} \quad h^2 h_l \cdots h_1 \leq Hhh_l \cdots h_1$$

$$\text{(Help3)} \quad Hhh_l \cdots h_2 f \leq Hhh_l \cdots h_2 g$$

for all $h : \mathbb{H}^{(l+1)}, h_l : \mathbb{H}^{(l)}, \dots, h_1 : \mathbb{H}^{(1)}$, and $f, g : \mathbb{H}^{(1)}$ with $f \leq g$.

Let $\mathbb{H}^{(l+2)}$ be the class of helpful functions on level $l + 2$. ■

This is a construction by recursion on the level l to produce $\mathbb{H}^{(l+1)} \subseteq \text{Ord}^{(l+1)}$. The comparison in (Help2, Help3) takes place in Ord' . Since the functions involved are in $\mathbb{I}\mathbb{M}$, this doesn't lead to difficulties.

For the case $l = 0$ you should read (Help3) with some care, because the sequence h, h_l, \dots, h_2 is empty. A function $H : \text{Ord}''$ is in $\mathbb{H}^{(2)}$ precisely when

$$(1) \quad Hh \in \mathbb{H}^{(1)} \quad (2) \quad h^2 \leq Hh \quad (3) \quad Hf \leq Hg$$

hold for all $f, g, h : \mathbb{H}^{(1)}$ with $f \leq g$.

In the next few results we develop some of the properties of the classes $\mathbb{H}^{(l+1)}$. Almost all of these are proved in section 7 of $[\lambda\text{H}]$.

We begin by generalizing (Help2).

11.20 LEMMA. If $H \in \mathbb{H}^{(l+2)}$, $h \in \mathbb{H}^{(l+1)}$, $h_l \in \mathbb{H}^{(l)}$, \dots , $h_1 \in \mathbb{H}^{(1)}$, then $(h\mathbf{h})^2 \leq Hh\mathbf{h}$ (where \mathbf{h} abbreviates $h_l \cdots h_1$).

This squaring property (Help2) is quite powerful and is a crucial component of the proof of many of the following results. The first of these generalizes part of Lemma 11.10.

11.21 LEMMA. For each $l < \omega$, the class $\mathbb{H}^{(l+1)}$ is smooth.

This has a useful consequence.

11.22 COROLLARY. For each $l < \omega$, $H \in \mathbb{H}^{(l+2)}$, $h \in \mathbb{H}^{(l+1)}$ we have $H^\alpha h \in \mathbb{H}^{(l+1)}$ for each $\alpha \in \text{Ord}$.

With these results we obtain higher level analogues of Lemma 11.18. This is Lemma 7.18 of [λH] together with an extra part (c).

11.23 LEMMA. Let $H : \mathbb{H}^{(l+2)}$, $h : \mathbb{H}^{(l+1)}$, $h_l : \mathbb{H}^{(l)}$, \dots , $h_1 : \mathbb{H}^{(1)}$. The the function

$$\alpha \mapsto H^\alpha h\mathbf{h}\zeta$$

is normal, and

$$(a) H^\alpha h\mathbf{h}\alpha \leq H^{\alpha+1} h\mathbf{h}0$$

$$(b) H^\lambda h\mathbf{h}\zeta = H^\lambda h\mathbf{h}0$$

$$(c) (\zeta < \nu = H^\nu h\mathbf{h}0) \iff (0 < \nu = H^\nu h\mathbf{h}\zeta)$$

hold for all ordinals α, ν, ζ and (additively) critical ordinal λ with $\zeta < \lambda$.

Let's look at the proof of part (c).

Consider any ordinal

$$\mu = H^\nu h\mathbf{h}\zeta$$

where $\nu \neq 0$. If $\nu = \alpha + 1$ then

$$\mu = H(H^\alpha h)\mathbf{h}\zeta$$

so that μ is a value of the helpful function $(H^\alpha h)\mathbf{h}$, and hence is a limit ordinal (and even critical). If ν is a limit ordinal, then μ is a supremum of such values, and hence is a limit ordinal. Note also that $\zeta < \mu$, so that two uses of part (b) gives

$$\mu = H^\nu h\mathbf{h}\eta$$

for each $\eta < \nu$. These observations lead to both implications of (c).

Why do we need this equivalence and that of Lemma 11.18(c)? In a moment we will define certain functions which look for fixed points

$$\nu = H^\nu h\mathbf{h}0$$

for given helpful functions H, h, \mathbf{h} . The equivalence (c) allows us to replace the argument 0 by any ordinal below ν (which will be critical). In particular, we may replace it by ω .

There is one final property of helpfulness that we need. This is another consequence of the squaring property (Help2).

11.24 LEMMA. For each $H : \mathbb{H}^{(2)}, h : \mathbb{H}^{(1)}$ the comparison

$$h^{2^\alpha} \leq H^\alpha h$$

holds for each ordinal α .

Shortly we will name certain particular helpful functions (on each level), but before that let's look at another possible development of this material.

The first observation is that a helpful function on some level is applied only to a helpful argument at the next level down (where each ordinal is viewed as helpful). We may set $\mathbb{H}^{(0)} = \text{Ord}$. In fact, we are not interested in the behaviour of a helpful function outside the helpful arguments. The next observation is that a helpful function applied to a helpful argument returns a helpful value. This is the condition (Help1). Thus we could define

$$\mathbb{H}^{(l+1)} = \text{those functions of type } \mathbb{H}^{(l)} \longrightarrow \mathbb{H}^{(l)}$$

which satisfy certain restriction. At the same time we can partially order each class $\mathbb{H}^{(l)}$. Since $\mathbb{H}^{(0)} = \text{Ord}$ we have the linear comparison on $\mathbb{H}^{(0)}$, and since $\mathbb{H}^{(1)} \subseteq \mathbb{IM}$ we have the pointwise comparison on $\mathbb{H}^{(1)}$. This idea lift all the way up the levels. Thus for $H, K \in \mathbb{H}^{(l+2)}$ we use

$$H \leq K \iff (\forall h \in \mathbb{H}^{(l)})[Hh \leq Kh]$$

to produce a pointwise comparison on $\mathbb{H}^{(l+2)}$. With this we find that

$$h^2 \leq Hh \quad H \text{ is monotone}$$

are rephrasings of (Help2) and (Help3). Furthermore, the pointwise supremum on $\mathbb{H}^{(l)}$ is the actual supremum. As a consequence of this we obtain higher level analogues of Lemma 11.7. Thus

$$H^\alpha \circ H^\beta = H^{\beta \circ \alpha} \quad (H^\beta)^\alpha = H^{\beta \times \alpha}$$

holds for each $H \in \mathbb{H}^{(l+2)}$ and ordinals α, β .

The problem with this approach is that it requires the definitions and properties to be developed in parallel, which can be a bit messy. Thus I chose to present it in the manner above.

The purpose of the development is to enable us to define certain functions and extract their properties.

11.25 DEFINITION. For each level $l < \omega$ let $[l] : \text{Ord}^{(l+2)}$ be the function given by

$$[l]h\mathbf{h} = \text{Fix}f \quad \text{where } f\alpha = h^\alpha h\mathbf{0} \text{ (for } \alpha \in \text{Ord})$$

for each compatible family h, \mathbf{h} of (helpful) functions. ■

Notice that we allow $[l]$ to consume only helpful arguments. This is because of the following.

11.26 THEOREM. For each $l < \omega$ the function $[l] : \text{Ord}^{(l+2)}$ is helpful.

Finally we can produce a sequence of ordinals.

11.27 DEFINITION. Let

$$\Delta[0] = \omega \quad \Delta[1] = \mathbf{Next}\omega \quad \Delta[l+2] = [l] \cdots [0] \mathbf{Next}\omega$$

for each $l < \omega$. ■

There are several obvious questions which spring to mind. What have the \mathbb{O} rd-based gadgets **Fix**, **Next**, $[l]$, $\Delta[l]$ got to do with the similarly named \mathbb{O} -base gadgets? What are the ordinals $\Delta[l]$? The final five sections of this survey are concerned with answering these questions, and setting the answers in a more general context.

12 Interpretation in \mathbb{O} rd

In this section we show that

$$\mathfrak{O}rd(\Delta[r]) = \Delta[r]$$

holds for each $r < \omega$. Think of what this means. The term $\vdash \Delta[r] : \mathcal{O}$ has an interpretation $\mathfrak{O}rd(\Delta[r])$ in \mathbb{O} which is passed by the mediating function

$$\mathbb{O} \xrightarrow{\mathfrak{O}rd(\cdot)} \mathbb{O}rd$$

to produced some ordinal. We wish to show $\Delta[r]$ is that ordinal. This description of $\mathfrak{O}rd(\Delta[r])$ is not the way we will calculate its value. By Lemma 10.7 we know that the term $\Delta[r]$ is canonical, so we will attempt to convert it into an equivalent algebraic term and then perform within $\mathbb{O}rd$ the iteration described by that template. Of course, this will not be entirely straight forward. For $r \geq 1$ the term $\Delta[r]$ contains names of some higher order functions on \mathbb{O} . Somehow we have to ‘interpret’ these names as higher order functions on $\mathbb{O}rd$. In particular, we have to show that for each $l < \omega$ the name $\vdash [l] : \mathcal{O}^{(l+2)}$ (as constructed in Definition 10.6) can be ‘interpreted’ by the function $[l] : \mathbb{O}rd^{(l+2)}$ (as constructed in Definition 11.25).

Our problem is to make precise the notion of an ‘interpretation’ in $\mathbb{O}rd$. We won’t do this in its full generality, we will deal only with enough terms for our immediate purpose. (There are some closed terms, such as $J_{\mathcal{N}}$, which do not have an ‘interpretation’ in $\mathbb{O}rd$.)

Let’s look at the evaluation of $\mathfrak{O}rd(\Delta[r])$ for $r = 0, 1, 2, \dots$ in turn.

Consider any limit structure \mathfrak{A} . A simple calculation shows that

$$\mathfrak{A}(\text{fin } \underline{m}) = A^m a$$

for each $m \in \mathbb{N}$, and hence

$$\mathfrak{A}(\omega) = \mathfrak{A}(\text{Lim fin}) = \mathcal{A}(m \mapsto A^m a)$$

holds. In particular

$$\mathfrak{O}rd(\Delta[0]) = \mathfrak{O}rd(\omega) = \bigvee \{m \mid m \in \mathbb{N}\} = \omega$$

as it should be.

How can we calculate $\mathfrak{O}rd(\Delta[1])$? We want to show that the function **Next** : $\mathbb{O}rd$ is the ‘interpretation’ of $\vdash \text{Next} : \mathcal{O}'$, but what can this mean?

Before we explain that it is useful to introduce a notation convention which is local to this section.

Throughout this section the problem is to match a term which inhabits $\mathcal{O}^{(r)}$ (for $r < \omega$) with some member of $\text{Ord}^{(r)}$. It is convenient to use the same letter for both the term and the member. However, we need to distinguish between these, so we place a \ulcorner over the term. Thus

$$\begin{aligned} \vdash \ulcorner \alpha \urcorner & \text{ will match } \alpha \in \text{Ord} \\ \vdash \ulcorner f \urcorner & \text{ will match } f \in \text{Ord}' \\ \vdash \ulcorner F \urcorner & \text{ will match } F \in \text{Ord}'' \\ & \vdots \end{aligned}$$

and so on. Of course, there are times when this convention is not needed, such as when showing that *Next* matches *Next*.

Using this convention we define a stratified matching relation.

12.1 DEFINITION. (0) For a term and ordinal

$$\ulcorner \alpha \urcorner \in \mathcal{C}_0 \quad \alpha \in \text{Ord}$$

we write

$$\ulcorner \alpha \urcorner \overset{0}{\sim} \alpha$$

and say $\ulcorner \alpha \urcorner$ matches α (at level 0) if $\mathfrak{Ord}(\ulcorner \alpha \urcorner) = \alpha$.

(1) For a term and function

$$\ulcorner f \urcorner \in \mathcal{C}_1 \quad f \in \text{Ord}'$$

we write

$$\ulcorner f \urcorner \overset{1}{\sim} f$$

and say $\ulcorner f \urcorner$ matches f (at level 1) if

$$\ulcorner \alpha \urcorner \overset{0}{\sim} \alpha \implies \ulcorner f \urcorner \ulcorner \alpha \urcorner \overset{0}{\sim} f\alpha$$

holds for all matching $\ulcorner \alpha \urcorner$ and α (at level 0).

($r + 2$) For a term and function

$$\vdash \ulcorner F \urcorner \in \mathcal{C}_{r+2} \quad F \in \text{Ord}'$$

we write

$$\ulcorner F \urcorner \overset{r+2}{\sim} F$$

and say $\ulcorner F \urcorner$ matches F (at level $r + 2$) if

$$\ulcorner f \urcorner \overset{r+1}{\sim} f \implies \ulcorner F \urcorner \ulcorner f \urcorner \overset{r+1}{\sim} Ff$$

holds for all matching $\ulcorner f \urcorner$ and f (at level $r + 1$). ■

We have isolated the clause for level 1 to help with some later proofs. Notice that we do not attempt to match all terms, only canonical terms on the appropriate level.

12.2 LEMMA. *Both the matchings*

$$\omega \overset{0}{\sim} \omega \quad \bar{S} \overset{1}{\sim} S$$

hold (where S is the successor function on $\mathbb{O}rd$).

Proof. The first of these is a restatement of the calculation above, and second is immediate. \blacksquare

This gives us simple examples of a 0-matching and a 1-matching. Shortly we will see an example of a 2-matching, and then examples of $(r + 2)$ -matchings for each r .

Using this notation the principal objective of this section is to show

$$\Delta[r] \overset{r}{\sim} \Delta[r]$$

for each $r < \omega$. We have already shown $\Delta[0] \overset{0}{\sim} \Delta[0]$ so now we start to move up the levels.

I said in Section 9 that Lemma 9.2 is enough to show that the ordinal operations

$$\mathfrak{O}rd[\bar{A}] \quad \mathfrak{O}rd[\bar{M}] \quad \mathfrak{O}rd[\bar{E}]$$

are just ordinal addition, multiplication, and exponentiation, but I quite deliberately didn't explain what this meant. Here is what it does mean.

12.3 LEMMA. *Suppose $\ulcorner \beta \urcorner \overset{0}{\sim} \beta$. Then*

$$(a) \quad \bar{A}^{\ulcorner \beta \urcorner} \overset{1}{\sim} \beta + \bullet \quad (m) \quad \bar{M}^{\ulcorner \beta \urcorner} \overset{1}{\sim} \beta \times \bullet \quad (e) \quad \bar{E}^{\ulcorner \beta \urcorner} \overset{1}{\sim} \beta^\bullet$$

hold.

Proof. The method of proof is obvious, but there are a couple of small points that so far I have carefully avoided. Notice that, by Corollary 9.4, each of $\bar{A}^{\ulcorner \beta \urcorner}, \bar{M}^{\ulcorner \beta \urcorner}, \bar{E}^{\ulcorner \beta \urcorner}$ is canonical, so that aspect is not a problem. Of course, we make use of Lemma 10.2.

(a) We show

$$\ulcorner \alpha \urcorner \overset{0}{\sim} \alpha \implies \bar{A}^{\ulcorner \beta \urcorner \ulcorner \alpha \urcorner} \overset{0}{\sim} \beta + \alpha$$

for all algebraic $\ulcorner \alpha \urcorner$. We proceed by induction on the structure of $\ulcorner \alpha \urcorner$.

For the base case, $\ulcorner \alpha \urcorner = \bar{0}$, we have

$$\bar{A}^{\ulcorner \beta \urcorner \bar{0}} \triangleright \ulcorner \beta \urcorner \quad \beta + 0 = \beta$$

which, since $\ulcorner \beta \urcorner \overset{0}{\sim} \beta$, gives the required result.

For the induction step $\alpha \mapsto \bar{S}^{\ulcorner \alpha \urcorner}$ we have

$$\bar{A}^{\ulcorner \beta \urcorner (\bar{S}^{\ulcorner \alpha \urcorner})} \bowtie \bar{S}(\bar{A}^{\ulcorner \beta \urcorner \ulcorner \alpha \urcorner}) \quad \bar{A}^{\ulcorner \beta \urcorner \ulcorner \alpha \urcorner} \overset{0}{\sim} \beta + \alpha$$

so the observation $\bar{S} \overset{1}{\sim} S$ leads to the required result.

For the induction leap consider $\ulcorner \alpha \urcorner = \ulcorner \mu \urcorner = \text{Lim } p$ where, for each $m \in \mathbb{N}$

$$pm \approx \ulcorner \alpha_m \urcorner \overset{0}{\sim} \alpha_m$$

and the induction hypothesis applies to each $\ulcorner \alpha_m \urcorner$. We need the ordinal μ such that $\ulcorner \mu \urcorner \overset{0}{\sim} \mu$. Thus

$$\begin{aligned} \mu &= \mathfrak{Ord}(\ulcorner \mu \urcorner) \\ &= \mathfrak{Ord}(\text{Lim } p) \\ &= \bigvee \{ \mathfrak{Ord}(p\underline{m}) \mid m \in \mathbb{N} \} \\ &= \bigvee \{ \mathfrak{Ord}(\ulcorner \alpha_m \urcorner) \mid m \in \mathbb{N} \} = \bigvee \{ \alpha_m \mid m \in \mathbb{N} \} \end{aligned}$$

as expected. However, we do *not* know that μ is a limit ordinal. We have put no restrictions on the behaviour of p , so it could happen that $\mu = \alpha_m$ for some m (if the set happens to have a maximum). We have to be aware of this possibility.

We have

$$\overline{\mathbf{A}}^{\ulcorner \beta \urcorner \ulcorner \mu \urcorner} \bowtie \text{Lim } q \quad \text{where} \quad q\underline{m} \approx \overline{\mathbf{A}}^{\ulcorner \beta \urcorner \ulcorner \alpha_m \urcorner}$$

(for each $m \in \mathbb{N}$). By the induction hypothesis we have

$$q\underline{m} \overset{0}{\sim} \beta + \alpha_m$$

so that

$$\begin{aligned} \mathfrak{Ord}(\overline{\mathbf{A}}^{\ulcorner \beta \urcorner \ulcorner \mu \urcorner}) &= \mathfrak{Ord}(\text{Lim } q) \\ &= \bigvee \{ \mathfrak{Ord}(q\underline{m}) \mid m \in \mathbb{N} \} \\ &= \bigvee \{ \beta + \alpha_m \mid m \in \mathbb{N} \} \\ &= \beta + \bigvee \{ \alpha_m \mid m \in \mathbb{N} \} = \beta + \mu \end{aligned}$$

as required. Notice that the fourth equality requires a few moment's thought.

(m) We show

$$\ulcorner \alpha \urcorner \overset{0}{\sim} \alpha \implies \overline{\mathbf{M}}^{\ulcorner \beta \urcorner \ulcorner \alpha \urcorner} \overset{0}{\sim} \beta \times \alpha$$

for all algebraic $\ulcorner \alpha \urcorner$. We proceed by induction on the structure of $\ulcorner \alpha \urcorner$, using part (a).

For the base case, $\ulcorner \alpha \urcorner = \overline{0}$, we have

$$\overline{\mathbf{M}}^{\ulcorner \beta \urcorner \overline{0}} \triangleright \overline{0} \quad \beta \times 0 = 0$$

which gives the required result.

For the induction step $\alpha \mapsto \overline{\mathbf{S}}^{\ulcorner \alpha \urcorner}$ we have

$$\overline{\mathbf{M}}^{\ulcorner \beta \urcorner (\overline{\mathbf{S}}^{\ulcorner \alpha \urcorner})} \bowtie \overline{\mathbf{A}}(\overline{\mathbf{M}}^{\ulcorner \beta \urcorner \ulcorner \alpha \urcorner})^{\ulcorner \beta \urcorner} \quad \beta \times (\alpha + 1) = \beta \times \alpha + \beta$$

and

$$\overline{\mathbf{M}}^{\ulcorner \beta \urcorner \ulcorner \alpha \urcorner} \overset{0}{\sim} \gamma \quad \text{where} \quad \gamma = \beta \times \alpha$$

by the induction hypothesis. Part (a) gives $\overline{\mathbf{A}}^{\ulcorner \gamma \urcorner} \overset{1}{\sim} \gamma + \bullet$ and hence

$$\overline{\mathbf{M}}^{\ulcorner \beta \urcorner (\overline{\mathbf{S}}^{\ulcorner \alpha \urcorner})} \overset{0}{\sim} \gamma + \beta = \beta \times (\alpha + 1)$$

as required.

For the induction leap consider $\ulcorner \alpha \urcorner = \ulcorner \mu \urcorner = \text{Lim } p$ where

$$p\underline{m} \approx \ulcorner \alpha_m \urcorner \overset{0}{\sim} \alpha_m \quad \ulcorner \mu \urcorner \overset{0}{\sim} \mu = \bigvee \{ \alpha_m \mid m \in \mathbb{N} \}$$

for each $m \in \mathbb{N}$ as in part (a). We know that either μ is a limit ordinal (if the set of all α_\bullet does not have a maximum) or $\mu = \alpha_m$ (if α_m is the maximum of that set).

Using Lemma 10.2, the induction hypothesis, and part (a) we have

$$M0^{\ulcorner \beta \urcorner \mu^{\lrcorner}} \bowtie \text{Lim } q \quad \text{where} \quad q \underline{m} \approx \overline{M}^{\ulcorner \beta \urcorner \alpha_m^{\lrcorner}} \overset{0}{\sim} \beta \times \alpha_m$$

(for each $m \in \mathbb{N}$). Thus

$$\mathfrak{Ord}(M0^{\ulcorner \beta \urcorner \mu^{\lrcorner}}) = \bigvee \{\beta \times \alpha_m \mid m \in \mathbb{N}\} = \beta \times \mu$$

as required. This second equality depends on the monotonicity of the function $\beta \times \bullet$.

(e) This is proved in a similar way making use of part (m). ■

With this we can produce our first example of a 2-matching. We match the term of Definition 10.5 with the function of Definition 11.11.

12.4 LEMMA. *The 2-matching*

$$\text{Fix} \overset{2}{\sim} \mathbf{Fix}$$

holds.

Proof. By Lemma 10.7 we have $\text{Fix} \in \mathcal{C}_2$, so it suffices to show

$$\text{Fix}^{\ulcorner f \urcorner \alpha^{\lrcorner}} \overset{0}{\sim} \mathbf{Fix} f \alpha$$

for matching pairs $\ulcorner f \urcorner \overset{1}{\sim} f$ and $\ulcorner \alpha \urcorner \overset{0}{\sim} \alpha$. By Definition 10.5 we have

$$\mathbf{Fix} f \alpha = f^\omega(S\alpha) = \bigvee \{f^m(S\alpha) \mid m \in \mathbb{N}\}$$

so we must match the term with this value. We have

$$\text{Fix}^{\ulcorner f \urcorner \alpha^{\lrcorner}} \triangleright (\mathcal{J}\omega\text{Lim})^{\ulcorner f \urcorner (\overline{S}^{\ulcorner \alpha \urcorner})} \triangleright \text{Lim } p \quad \text{where} \quad p \underline{m} \approx \ulcorner f^m \urcorner (\overline{S}^{\ulcorner \alpha \urcorner})$$

for each $m \in \mathbb{N}$. Since $\ulcorner f \urcorner \overset{1}{\sim} f$ we have

$$\ulcorner \zeta \urcorner \overset{0}{\sim} \zeta \implies \ulcorner f \urcorner \ulcorner \zeta \urcorner \overset{0}{\sim} f \zeta$$

which, by a simple induction, gives

$$\ulcorner f^m \urcorner (\overline{S}^{\ulcorner \alpha \urcorner}) \overset{0}{\sim} f^m(S\alpha)$$

for each $m \in \mathbb{N}$. With this we have

$$\begin{aligned} \mathfrak{Ord}(\text{Fix}^{\ulcorner f \urcorner \alpha^{\lrcorner}}) &= \mathfrak{Ord}(\text{Lim } p) \\ &= \bigvee \{\mathfrak{Ord}(p \underline{m}) \mid m \in \mathbb{N}\} \\ &= \bigvee \{f^m(S\alpha) \mid m \in \mathbb{N}\} = \mathbf{Fix} f \alpha \end{aligned}$$

as required. ■

It has taken longer that we might have thought, but we can now match $\Delta[1]$.

12.5 LEMMA. *Both the matchings*

$$\mathbf{Next} \overset{1}{\sim} \mathbf{Next} \quad \Delta[1] \overset{0}{\sim} \Delta[1]$$

hold.

Proof. By Lemmas 12.4, 12.3, and 12.2 we have

$$\mathbf{Next} = \text{Fix}(\bar{E}\omega) \overset{0}{\sim} \mathbf{Fix} \omega^\bullet = \mathbf{Next}$$

and hence

$$\Delta[1] = \mathbf{Next} \omega \overset{0}{\sim} \mathbf{Next} \omega = \Delta[1]$$

as required. ■

It is now clear how we should verify $\Delta[2] \overset{0}{\sim} \Delta[2]$. If we can show $[0] \overset{2}{\sim} [0]$ then Lemmas 12.5 and 12.2 give

$$\Delta[2] = [0]\mathbf{Next} \omega \overset{0}{\sim} [0]\mathbf{Next} \omega = \Delta[2]$$

as we want. Unfortunately, it's not that simple.

How might we show $[0] \overset{2}{\sim} [0]$? Consider a matching $\ulcorner h \urcorner \overset{1}{\sim} h$. We have

$$[0]\ulcorner h \urcorner \approx \text{Fix}\ulcorner f \urcorner \quad [0]h = \mathbf{Fix}f$$

where

$$\ulcorner f \urcorner = \lambda\alpha : \mathcal{O}. (\text{JaLim})\ulcorner h \urcorner \bar{0} \quad f\alpha = h^\alpha 0$$

give the auxiliary components. It suffices to show $\ulcorner f \urcorner \overset{1}{\sim} f$. To do this we will have to consider

$$\ulcorner \alpha \urcorner = \ulcorner \mu \urcorner = \text{Lim } p \quad \text{where } p\bar{m} \approx \ulcorner \alpha_m \urcorner \overset{0}{\sim} \alpha_m$$

for each $m \in \mathbb{N}$. We have

$$\ulcorner \mu \urcorner \overset{0}{\sim} \mu = \bigvee \{\alpha_m \mid m \in \mathbb{N}\}$$

(as in the proof of Lemma 12.4). A simple calculation gives

$$\ulcorner f \urcorner \ulcorner \mu \urcorner \overset{0}{\sim} \bigvee \{f\alpha_m \mid m \in \mathbb{N}\}$$

so that

$$f\mu = h^\mu = \bigvee \{h^{\alpha_m} \mid m \in \mathbb{N}\} = \bigvee \{f\alpha_m \mid m \in \mathbb{N}\}$$

is the crux of the proof. Unfortunately, this central equality is *not* true for an arbitrary $h : \text{Ord}'$. We need $\alpha \mapsto h^\alpha \zeta$ to be monotone for each ζ .

To deal with this we refine the matching relation to ensure that the functions that arise have appropriate properties. For our purposes we can use helpful functions.

12.6 DEFINITION. (0) For a term and ordinal

$$\ulcorner \alpha \urcorner \in \mathcal{C}_0 \quad \alpha \in \text{Ord}$$

we write

$$\ulcorner \alpha \urcorner \stackrel{0}{\simeq} \alpha$$

and say $\ulcorner \alpha \urcorner$ helpfully matches α (at level 0) if $\mathfrak{Drd}(\ulcorner \alpha \urcorner) = \alpha$.

(1) For a term and function

$$\ulcorner h \urcorner \in \mathcal{C}_1 \quad h \in \mathbb{H}^{(1)}$$

we write

$$\ulcorner h \urcorner \stackrel{1}{\simeq} h$$

and say $\ulcorner h \urcorner$ helpfully matches h (at level 1) if

$$\ulcorner \alpha \urcorner \stackrel{0}{\simeq} \alpha \implies \ulcorner h \urcorner \ulcorner \alpha \urcorner \stackrel{0}{\simeq} h\alpha$$

holds for all helpfully matching $\ulcorner \alpha \urcorner$ and α (at level 0).

($r + 2$) For a term and function

$$\ulcorner F \urcorner \in \mathcal{C}_{r+2} \quad F \in \mathbb{H}^{(r+2)}$$

we write

$$\ulcorner F \urcorner \stackrel{r+2}{\simeq} F$$

and say $\ulcorner F \urcorner$ helpfully matches F (at level $r + 2$) if

$$\ulcorner f \urcorner \stackrel{r+1}{\simeq} f \implies \ulcorner F \urcorner \ulcorner f \urcorner \stackrel{r+1}{\simeq} Ff$$

holds for all helpfully matching $\ulcorner f \urcorner$ and f (at level $r + 1$). ■

This matching relation $\overset{\bullet}{\simeq}$ is like the earlier one $\overset{\bullet}{\sim}$ except we now require that the matched function is helpful (at the appropriate level). This will enable us to proceed with several inductions. Notice that $\overset{0}{\simeq}$ and $\overset{0}{\sim}$ are the same relation. Notice also that

$$\ulcorner h \urcorner \stackrel{1}{\simeq} h \iff \ulcorner h \urcorner \stackrel{1}{\sim} h \text{ and } h \in \mathbb{H}^{(1)}$$

but similar implications do not hold at higher levels.

We already have an example of $\overset{1}{\simeq}$. By Lemma 11.13 the function **Next** is helpful, so Lemma 12.5 give the following.

12.7 LEMMA. *The helpful matching*

$$\text{Next} \stackrel{1}{\simeq} \mathbf{Next}$$

holds.

We use this refined matching in the crucial observation.

12.8 LEMMA. *The implications*

$$(0) \quad \left. \begin{array}{l} \ulcorner h \urcorner \overset{1}{\simeq} h \\ \ulcorner \zeta \urcorner \overset{0}{\simeq} \zeta \\ \ulcorner \alpha \urcorner \overset{0}{\simeq} \alpha \end{array} \right\} \implies (\mathbf{J}^{\ulcorner \alpha \urcorner} \mathbf{Lim}) \ulcorner h \urcorner \ulcorner \zeta \urcorner \overset{0}{\simeq} h^\alpha \zeta$$

$$(l+1) \quad \left. \begin{array}{l} \ulcorner H \urcorner \overset{l+2}{\simeq} H \\ \ulcorner h \urcorner \overset{l+1}{\simeq} h \\ \ulcorner \alpha \urcorner \overset{0}{\simeq} \alpha \end{array} \right\} \implies (\mathbf{J}^{(l+1)\ulcorner \alpha \urcorner} \mathbf{Lim}^{(l+1)}) \ulcorner H \urcorner \ulcorner h \urcorner \overset{l+1}{\simeq} H^\alpha h$$

hold (for all $l < \omega$).

Proof. As we will see, with a little care these two clauses could be combined into a single implication, but it is clearer if we deal with them separately. The proof of (0) is a simplified version of the proof of $(l+1)$ and, in essence, is a corrected version of the false proof just before Definition 12.6. In view of this we may concentrate on the proof on $(l+1)$.

We keep

$$\ulcorner H \urcorner \overset{l+2}{\simeq} H \quad \ulcorner h \urcorner \overset{l+1}{\simeq} h$$

fixed throughout. Consider any $\ulcorner \alpha \urcorner \overset{0}{\simeq} \alpha$. Since $\ulcorner \alpha \urcorner \in \mathcal{C}_0$ we have

$$(\mathbf{J}^{(l+1)\ulcorner \alpha \urcorner} \mathbf{Lim}^{(l+1)}) \in \mathcal{C}_{l+3}$$

by Lemma 9.3. Thus

$$(\mathbf{J}^{(l+1)\ulcorner \alpha \urcorner} \mathbf{Lim}^{(l+1)}) \ulcorner H \urcorner \ulcorner h \urcorner \in \mathcal{C}_{l+1}$$

since $\ulcorner H \urcorner \in \mathcal{C}_{l+2}$ and $\ulcorner h \urcorner \in \mathcal{C}_{l+1}$. We have $H \in \mathbb{H}^{(l+2)}$ and $h \in \mathbb{H}^{(l+1)}$ so that $H^\alpha h \in \mathbb{H}^{(l+1)}$ by Corollary 11.22. This sorts out the bookkeeping. We now proceed by induction over the structure of $\ulcorner \alpha \urcorner$ (for algebraic $\ulcorner \alpha \urcorner$).

The base case, $\ulcorner \alpha \urcorner = \bar{0}$, is immediate, and the induction step, $\ulcorner \alpha \urcorner \mapsto \bar{\mathbf{S}}\ulcorner \alpha \urcorner$ is straight forward. We look at the case $\ulcorner \alpha \urcorner = \ulcorner \mu \urcorner = \mathbf{Lim} p$ where

$$p\bar{m} \approx \ulcorner \alpha_m \urcorner \overset{0}{\simeq} \alpha_m$$

for each $m \in \mathbb{N}$. (Here we are using the internal equality \approx informally.) As in the proof of Lemma 12.4 we have

$$\ulcorner \mu \urcorner \overset{0}{\simeq} \mu = \bigvee \{\alpha_m \mid m \in \mathbb{N}\}$$

where either μ is a limit ordinal or $\mu = \alpha_m$ for some m .

To show

$$(\mathbf{J}^{(l+1)\ulcorner \alpha \urcorner} \mathbf{Lim}^{(l+1)}) \ulcorner H \urcorner \ulcorner h \urcorner \overset{l+1}{\simeq} H^\alpha h$$

consider an arbitrary selection

$$\ulcorner h_l \urcorner \overset{l}{\simeq} h_l, \dots, \ulcorner h_1 \urcorner \overset{1}{\simeq} h_l, \ulcorner \zeta \urcorner \overset{0}{\simeq} \zeta$$

so that

$$(\mathbf{J}^{(l+1)\ulcorner \alpha \urcorner} \mathbf{Lim}^{(l+1)}) \ulcorner H \urcorner \ulcorner h \urcorner \ulcorner h \urcorner \ulcorner \zeta \urcorner \overset{0}{\simeq} H^\alpha h h \zeta$$

will give the required result. Here we have use the obvious notation to condense the list of auxiliaries.

By Lemma 11.23(a) the function $\alpha \mapsto H^\alpha h \mathbf{h} \zeta$ is normal, and hence

$$H^\mu h \mathbf{h} \zeta = \bigvee \{H^{\alpha_m} h \mathbf{h} \zeta \mid m \in \mathbb{N}\}$$

(where the two possibilities for μ should be considered).

Now

$$(J^{(l+1)} \ulcorner \alpha \urcorner \text{Lim}^{(l+1)}) \ulcorner H \urcorner \ulcorner h \urcorner = (J^{(l+1)} (\text{Lim } p) \text{Lim}^{(l+1)}) \ulcorner H \urcorner \ulcorner h \urcorner \approx q$$

where, by the induction hypothesis

$$qm \approx (J^{(l+1)} \ulcorner \alpha_m \urcorner \text{Lim}^{(l+1)}) \ulcorner H \urcorner \ulcorner h \urcorner \stackrel{l+1}{\cong} H^{\alpha_m} h$$

for each $m \in \mathbb{N}$. Thus

$$\mathfrak{Drd}((J^{(l+1)} \ulcorner \alpha \urcorner \text{Lim}^{(l+1)}) \ulcorner H \urcorner \ulcorner h \urcorner \ulcorner \mathbf{h} \urcorner \ulcorner \zeta \urcorner) = \bigvee \{H^{\alpha_m} h \mathbf{h} \zeta \mid m \in \mathbb{N}\} \stackrel{0}{\cong} H^\mu h \mathbf{h} \zeta$$

as required. ■

This quickly gives us the main result of this section.

12.9 THEOREM. *The matching*

$$[l] \stackrel{l+2}{\cong} [l]$$

holds for each $l < \omega$.

Proof. As in the proof of Lemma 12.8 it is convenient to prove

$$[0] \stackrel{2}{\cong} [0] \quad [l+1] \stackrel{l+3}{\cong} [l]$$

separately. Also, we may concentrate on the second of these.

By Lemma 10.7 and Theorem 11.26 we have $[l+1] \in \mathcal{C}_{l+3}$ and $[l+1] \in \mathbb{H}^{(l+3)}$. Thus it suffices to show

$$[l+1] \ulcorner H \urcorner \ulcorner h \urcorner \ulcorner \mathbf{h} \urcorner \ulcorner \zeta \urcorner \stackrel{0}{\sim} [l+1] H h \mathbf{h} \zeta$$

for each selection

$$\ulcorner H \urcorner \stackrel{l+2}{\cong} H, \ulcorner h \urcorner \stackrel{l+1}{\cong} h, \ulcorner h_l \urcorner \stackrel{l}{\cong} h_l, \dots, \ulcorner h_1 \urcorner \stackrel{1}{\cong} h_1, \ulcorner \zeta \urcorner \stackrel{0}{\cong} \zeta$$

of matchings (where the obvious condensing notation has been used).

We have

$$[l+1] \ulcorner H \urcorner \ulcorner h \urcorner \ulcorner \mathbf{h} \urcorner \ulcorner \zeta \urcorner \approx \text{Fix} \ulcorner f \urcorner \ulcorner \zeta \urcorner \quad [l+1] H h \mathbf{h} \zeta = \mathbf{Fix} f \zeta$$

where

$$\ulcorner f \urcorner = \lambda \alpha : \mathcal{O} . (J^{(l+1)} \ulcorner \alpha \urcorner \text{Lim}^{(l+1)}) \ulcorner H \urcorner \ulcorner h \urcorner \bar{0} \quad f \alpha = H^\alpha h \mathbf{h} 0$$

give the auxiliaries. By Lemma 12.4 a matching

$$\ulcorner f \urcorner \stackrel{1}{\sim} f$$

will suffice. (Note that here we do not use the helpful matching relation.) Certainly $\ulcorner f \urcorner \in \mathcal{C}_1$. Also, for $\ulcorner \alpha \urcorner \overset{0}{\sim} \alpha$, Lemma 12.8 gives

$$\ulcorner f \urcorner \ulcorner \alpha \urcorner \overset{1}{\sim} f\alpha$$

and hence $\ulcorner f \urcorner \overset{1}{\sim} f$, as required. ■

This gives

$$[l+r] \cdots [r] \overset{r+2}{\simeq} [l+r] \cdots [r]$$

(for each $l, r \in \mathbb{N}$) and hence, since $\text{Next} \overset{1}{\simeq} \mathbf{Next}$ and $\omega \overset{0}{\simeq} \omega$, we obtain the result we have been looking for.

12.10 COROLLARY. *For each $r < \omega$ the matching*

$$\Delta[r] \overset{0}{\sim} \Delta[r]$$

holds.

It is clear that the methods of this section have been cobbled together merely to prove this last result. I do not know if the methods use can be extended to obtain more general notions of ‘interpretation’ of parts of $\lambda\mathbf{H}$ in Ord .

13 Veblen hierarchies

How might we enumerate a long initial stretch of countable ordinals? Furthermore, how might we produce a notation for each member of such a long stretch?

In the naming process we are allowed to use certain predetermined gadgets. For instance, and natural number may be used. We may also use ω , the least infinite ordinal. More importantly, we may also use the name of any ordinal already generated.

The crucial trick is the generalized Cantor normal form.

13.1 THEOREM. *Let θ be a critical ordinal, and suppose we can name all critical ordinals $\epsilon < \theta$. Then we can name all ordinals $\zeta < \theta$.*

Proof. We set up a recursive procedure for naming these ordinals. We can, of course, name the finite ordinals, and ω is part of our supply of names. We may use addition, multiplication (on the right by a natural), and exponentiation (to base ω).

Consider any ordinal $\zeta < \theta$. By the remarks above we may assume that $\zeta \neq 0$.

Since

$$\zeta < (\zeta + 1) \leq \omega^{\zeta+1}$$

we have

$$\zeta < \omega^\eta$$

for at least one ordinal η . Let η be the smallest example of such an ordinal. In particular, we have

$$\omega^\alpha \leq \zeta$$

for each $\alpha < \eta$. This η can not be a limit ordinal. (For if it is then

$$\omega^\eta = \bigvee \{\omega^\alpha \mid \alpha < \eta\} \leq \zeta$$

which is not so.) This η can not be 0. (For otherwise

$$\zeta < \omega^0 = 1$$

which is not so.) Thus η is a successor ordinal.

Thus we have

$$\omega^\alpha \leq \zeta < \omega^{\alpha+1} \tag{1}$$

for at least one ordinal.

There can not be two such ordinals. For suppose there are, say $\alpha < \beta$. Then $\alpha + 1 \leq \beta$ so that

$$\omega^\beta \leq \zeta < \omega^{\alpha+1} \leq \omega^\beta$$

which is impossible.

The unique ordinal α is the first component in our name for ζ .

We have

$$\alpha \leq \omega^\alpha \leq \zeta$$

for the α . If $\alpha = \zeta$ then ζ is critical with $\zeta < \theta$, and so ζ can be named. Thus we may suppose that $\alpha < \zeta$, and hence α can be named by the recursive procedure.

Consider the ascending chain

$$0 < \omega^\alpha < \omega^\alpha \cdot 2 < \omega^\alpha \cdot 3 < \dots < \omega^\alpha \cdot m < \dots$$

for $m \in \mathbb{N}$. If all of these lie below ζ then

$$\omega^{\alpha+1} = \bigvee \{\omega^\alpha \cdot m \mid m \in \mathbb{N}\} \leq \zeta$$

which is not so. Thus we have

$$\omega^\alpha \cdot m \leq \zeta < \omega^\alpha \cdot (m + 1) \tag{2}$$

for some unique $m \in \mathbb{N}$. This is the second component in our name for ζ .

Since

$$\omega^\alpha \cdot (m + 1) = \omega^\alpha \cdot m + \omega^\alpha$$

we have

$$\omega^\alpha \cdot m \leq \zeta < \omega^\alpha \cdot m + \eta$$

for at least one ordinal. Let η be the smallest example of such an ordinal. By a slight modification of the previous argument we see that η is non-zero and not a limit ordinal. Thus we have

$$\omega^\alpha \cdot m + \beta \leq \zeta < \omega^\alpha \cdot m + \beta + 1 \tag{3}$$

for some unique ordinal $\beta < \omega^\alpha$. This unique ordinal β is the third component in our name for ζ .

Since

$$\omega^\alpha \cdot m + \beta \quad \omega^\alpha \cdot m + \beta + 1$$

are consecutive ordinals we have

$$\zeta = \omega^\alpha \cdot m + \beta \tag{4}$$

for these uniquely determined components. We have seen above that we may suppose $\alpha < \zeta$. Also

$$\beta < \omega^\alpha \leq \zeta$$

so both α and β are strictly below ζ . Thus we may invoke the recursive procedure. ■

This shows that to name ordinals it is sufficient to generate a long enough stretch of critical ordinals, the fixed points of the function ω^\bullet

$$\alpha \longmapsto \omega^\alpha$$

exponentiation to base ω . We know that

$$\mathbf{Next} = \mathbf{Fix}\omega^\bullet$$

outputs the next critical strictly beyond its supplied input. Thus

$$\alpha \longmapsto \mathbf{Next}^{1+\alpha}0$$

is an enumeration of ‘all’ critical ordinals. Thus once we have named an ordinal α we may use the critical

$$\epsilon_\alpha = \mathbf{Next}^{1+\alpha}0$$

to name a whole new stretch of ordinals. With these we can generate more critical ordinals and so name a longer stretch of ordinals. These give us even more criticals, and so on.

Of course, there is a problem.

We always have

$$\alpha \leq \epsilon_\alpha = \mathbf{Next}^{1+\alpha}0$$

but at some point we meet a barrier ordinal, a fixed point of the enumerating function. For \mathbf{Next} the first of these is

$$\epsilon_{\epsilon_{\dots}}$$

the least critical ordinal which is its own index.

What do we do next?

One possibility is to invent a name for this barrier ordinal, and then allow this in the naming process. Thus if ϵ is the barrier ordinal then

$$\alpha \longmapsto \mathbf{Next}^{1+\alpha}\epsilon$$

enumerates many more critical ordinals until we meet the next barrier ordinal.

We want to iterate this process. Enumerate up to a barrier ordinal, invent a name for that, and then continue further. Of course, we want a systematic method of carrying out this process. That is what Veblen provided in [26].

Suppose we are allowed to use the function

$$f : \text{Ord} \longrightarrow \text{Ord}$$

in the naming process. Let us think of this as the base function. Above this is the function ω^\bullet , but we need to generalize that process. This base function f will be normal, or even fruitful. Thus f has many fixed points, and each one of these is critical.

We know that

$$\mathbf{Fix} f$$

outputs the next fixed point of f strictly beyond its supplied input. Thus

$$\alpha \longmapsto (\mathbf{Fix} f)^{1+\alpha} 0$$

is an enumeration of certain critical ordinals, and we are allowed to use these in the naming process. Of course, eventually this enumeration hits a barrier ordinal

$$\nu = (\mathbf{Fix} f)^{1+\nu} 0$$

and then we need something new. Veblen's idea is to use the enumeration of these barrier ordinals as a new base function.

We start from any normal (or even fruitful) function $f : \text{Ord}'$. We generate a whole hierarchy of base functions. Thus we set

$$\begin{aligned} \phi[f]0 &= f \\ \phi[f](\alpha + 1) &= \text{enumeration of fixed points of } \phi[f]\alpha \\ \phi[f]\lambda &= \text{enumeration of common fixed points of } \phi[f]\alpha \text{ for all } \alpha < \lambda \end{aligned}$$

for each ordinal α and limit ordinal λ . Of course, we need to check that each is normal (or fruitful). We see how to do that in a moment. Also, at some point this hierarchy will run out of steam. There is a critical ordinal ν with

$$\nu = \phi[f]\nu 0$$

and the hierarchy can not name ordinals beyond this. For the standard case where $f = \omega^\bullet$ this ordinal is Γ_0 . Veblen suggested how we can get larger ordinal, and we will come to that shortly.

Let's reorganize Veblen's idea in terms of the gadgetry we now have.

By Lemma 11.11 we have two function

$$\mathbf{Fix} : \text{Fruit} \longrightarrow \text{Help} \quad \mathbf{Enm} : \text{Help} \longrightarrow \text{Fruit}$$

given by

$$\mathbf{Fix} f \zeta = \text{least } \nu \text{ with } \zeta < \nu = f\nu \quad \mathbf{Enm} h\alpha = h^{1+\alpha} 0$$

for each $f \in \text{Fruit}$, $h \in \text{Help}$ and ordinals α, ζ . Of course, \mathbf{Fix} is the function used above. Recall also that $\mathbf{Enm} h$ is normal.

These two functions can be composed both ways to produce two more functions. It does take too much effort to show that

$$\mathbf{Fix} \circ \mathbf{Enm} = [0]$$

but it is the other composite we want now.

13.2 DEFINITION. Let

$$\mathbf{Veb} = \mathbf{Enm} \circ \mathbf{Fix}$$

to produce a function $\mathbf{Veb} : \text{Fruit} \longrightarrow \text{Fruit}$. ■

In other words we have

$$\mathbf{Veb}f\alpha = h^{1+\alpha}0 \quad \text{where } h = \mathbf{Fix}f$$

for each fruitful function f and ordinal α . Thus $\mathbf{Veb}f$ enumerates the fixed points of the function f , and hence

$$\phi[f](\alpha + 1) = \mathbf{Veb}(\phi[f]\alpha)$$

is the way we get from one level of the Veblen hierarchy to the next.

This function

$$\mathbf{Veb} : \mathbb{Fruit} \longrightarrow \mathbb{Fruit}$$

can be iterated through the ordinals, thus

$$\mathbf{Veb}^0 f = f \quad \mathbf{Veb}^{\alpha+1} f = \mathbf{Veb}(\mathbf{Veb}^\alpha f) \quad \mathbf{Veb}^\lambda f = \bigvee \{ \mathbf{Veb}^\alpha f \mid \alpha < \lambda \}$$

for each fruitful function f , each ordinal α and each limit ordinal λ . Note that by Lemma 11.10 each one of these functions is fruitful. More importantly, even if the start function f is normal the limit level functions $\mathbf{Veb}^\lambda f$ are only fruitful. This is the main reason why fruitful functions are more convenient than normal functions.

What is happening at limit levels?

As a consequence of Lemma 11.14 we have an ascending chain

$$f \leq \mathbf{Veb}f \leq \mathbf{Veb}^2 f \leq \cdots \leq \mathbf{Veb}^\alpha f \leq \cdots \quad (\alpha < \Omega)$$

of fruitful functions. This is related to the Veblen hierarchy $\phi[f]$. The limit levels of the Veblen hierarchy require the common fixed points of a family of functions. This can be simplified. Also as a consequence of Lemma 11.14 we have the following.

13.3 LEMMA. *For each directed family \mathcal{F} of fruitful functions, the common fixed points of the members of \mathcal{F} are precisely the fixed points of $\bigvee \mathcal{F}$.*

This shows that

$$\phi[f]\lambda = \mathbf{Veb}(\bigvee \{ \phi[f]\alpha \mid \alpha < \lambda \})$$

for each limit ordinal λ , and so we obtain the following.

13.4 LEMMA. *For each fruitful function f we have*

$$\phi[f](1 + \alpha) = \mathbf{Veb}^{\alpha+1} f$$

for each ordinal α .

Proof. This follows by a routine base-step-jump induction over α making use of the previous observation at limit levels. For instance, for each limit ordinal λ we have

$$\begin{aligned} \phi[f]\lambda &= \mathbf{Veb}(\bigvee \{ \phi[f]\alpha \mid \alpha < \lambda \}) \\ &= \mathbf{Veb}(\bigvee \{ \phi[f](1 + \alpha) \mid \alpha < \lambda \}) \\ &= \mathbf{Veb}(\mathbf{Veb}^\lambda f) &= \mathbf{Veb}^{\lambda+1} f \end{aligned}$$

as required. The second equality holds since $\phi[f]0 = f$ and this can be omitted from the ascending chain. \blacksquare

The two hierarchies

$$\alpha \longmapsto \phi[f]\alpha \quad \alpha \longmapsto \mathbf{Veb}^\alpha f$$

interlace but are not exactly the same. It turns out that the right hand version is a more convenient tool. We can also convert this into an even neater form.

We have

$$\mathbf{Fix} \circ \mathbf{Veb} = \mathbf{Fix} \circ \mathbf{Enm} \circ \mathbf{Fix} = [0] \circ \mathbf{Fix}$$

and this equality can be generalized.

13.5 LEMMA. *We have*

$$\mathbf{Fix} \circ \mathbf{Veb}^\alpha = [0]^\alpha \circ \mathbf{Fix}$$

for each ordinal α .

Proof. This follows by induction on α . The base case is trivial, the induction step follows by the observation above, and the induction leap to a limit ordinal follows by the Scott continuity of \mathbf{Fix} . \blacksquare

This result gives us a ‘helpful’ description of the Veblen hierarchy.

13.6 THEOREM. *For each fruitful function f and ordinal α we have*

$$\phi[f](1 + \alpha) = \mathbf{Veb}^{\alpha+1} f = \mathbf{Enm}([0]^\alpha h)$$

where $h = \mathbf{Fix} f$.

Proof. Using Lemma 13.4 it suffices to verify the second equality. For that we have

$$\begin{aligned} \mathbf{Veb}^{\alpha+1} f &= (\mathbf{Veb} \circ \mathbf{Veb}^\alpha) f \\ &= (\mathbf{Enm} \circ \mathbf{Fix} \circ \mathbf{Veb}^\alpha) f \\ &= (\mathbf{Enm} \circ [0]^\alpha \circ \mathbf{Fix}) f = (\mathbf{Enm} \circ [0]^\alpha) h \end{aligned}$$

for the required result \blacksquare

This shows that for each base function f the Veblen hierarchy is essentially the indexed family of functions

$$\alpha \longmapsto [0]^\alpha h$$

where $h = \mathbf{Fix} f$, and we are beginning to see why the helpful functions are nice to us.

We can now consider the problem of what to do when the Veblen hierarchy $\phi[f]$ runs out of steam. In other words when the least ordinal ν with

$$\nu = \phi[f]\nu = [0]^\nu h$$

is met. Veblen’s idea was to repeat the Veblen hierarchy on a new base function, and this function should be some kind of diagonal limit through the previous hierarchy. It turns out that $\uparrow f$ given by

$$\uparrow f \alpha = \phi[f](1 + \alpha) = [0]^\alpha h$$

is the appropriate function.

To generate $\phi[\uparrow f]$ in helpful form we need to know what $\mathbf{Fix}(\uparrow f)$ is.

13.7 LEMMA. For each fruitful function f we have

$$\mathbf{Fix}(\uparrow f) = [1][0]h$$

where $h = *\mathbf{Fix}*f$.

Proof. Remembering the characteristic properties of \mathbf{Fix} and $[1]$ for each ordinal ζ we have

$$\mathbf{Fix}(\uparrow f)\zeta = (\text{least } \nu \text{ with } \zeta < \nu = \uparrow f\nu = [0]^\nu h0) = [1][0]h\zeta$$

for the required result. ■

This shows that in helpful form the first two Veblen hierarchies are

$$\begin{aligned} \alpha &\longmapsto ([0]^\alpha h) \\ \alpha &\longmapsto ([0]^\alpha ([1][0]h)) \end{aligned}$$

followed by

$$\begin{aligned} \alpha &\longmapsto ([0]^\alpha (([1][0])^2 h)) \\ \alpha &\longmapsto ([0]^\alpha (([1][0])^3 h)) \\ &\quad \vdots \\ \alpha &\longmapsto ([0]^\alpha (([1][0])^\omega h)) \\ &\quad \vdots \\ \alpha &\longmapsto ([0]^\alpha (([1][0])^\beta h)) \\ &\quad \vdots \end{aligned}$$

for a sufficient stretch of ordinals β . We can begin to see what we have to do.

Of course, all this needs to be organized, and Veblen does attempt an explanation of this. However, I must say I have never understood his description.

We can now see a much clearer explanation.

Consider all the helpful functions $H \in \mathbb{H}\text{elp}^{(2)}$ that can be built from legal compounds of $[0]$ and $[1]$. To produce these we can use composition, application (of a function in $\mathbb{H}\text{elp}^{(3)}$ to a function in $\mathbb{H}\text{elp}^{(2)}$), and ordinal iteration of such functions. We also allow the identity function.

For each such function we have a hierarchy

$$\alpha \longmapsto [0]^\alpha Hh$$

where

$$H = ([1][0])^\beta$$

gives the examples above.

This collection of hierarchies needs to be organized and we will see how to do that in the next two sections. In [22] Schütte did give an organization, of sorts, but it doesn't really explain what is going on. There is a description of this in [Sch], for what it's worth.

We need to organize all the functions in $\mathbb{H}\text{elp}^{(2)}$ that can be obtained from $[0]$ and $[1]$. But why stop at these two? It is almost as easy to organize all the functions in $\mathbb{H}\text{elp}^{(2)}$ that can be obtain from

$$[0], [1], [2], \dots, [l], \dots$$

our collection of special helpful functions.

14 Uncountable ordinals as templates

For each level l we have a special helpful function, a fixed point extractor.

$$[l] : \mathbb{H}^{(l+2)} \subseteq \text{Ord}^{(l+2)}$$

Using ordinal iterates, legal composition, and application these generate a vast collection of helpful functions. We need a method of naming these functions.

For each list

$$\alpha(r), \dots, \alpha(1), \alpha(0)$$

of (countable) ordinals let

$$((\alpha(r), \dots, \alpha(1), \alpha(0)))_l$$

be the composite

$$\left(\left(\dots \left(([l+r]^{\alpha(r)} [l+r-1])^{\alpha(r-1)} [l+r-2] \right)^{\alpha(r-2)} \dots \right)^{\alpha(1)} [l] \right)^{\alpha(0)}$$

to obtain a member of $\mathbb{H}^{(l+2)}$. This construction gives us a rather pictorial idea of the helpful function produced. Of course, almost certainly it will be better to have a recursive construction of these functions.

In the following we write \perp for the empty list and β for an arbitrary list of (countable) ordinals.

14.1 DEFINITION. For each level l we set

$$((\perp))_l = \mathbf{id}^{(l+2)} \quad ((\beta\alpha))_l = (((\beta))_{l+1} [l])^\alpha$$

for each list β of ordinals and ordinal α . ■

This is not all the helpful functions we can generate. For each list

$$\beta_s, \dots, \beta_1$$

of lists of ordinals the composite

$$((\beta_s))_l \circ \dots \circ ((\beta_1))_l$$

is a member of $\mathbb{H}^{(l+2)}$.

We need some method of describing these gadgets. It is clear that each one is determine by a network of countable ordinals, sometimes a rather intricate network of ordinals. How can we notate such networks?

The method we use was devised by Bachmann. We use certain uncountable ordinals! That may seem a bit weird but once we see that explanation it makes perfect sense. It isn't their ordinal properties that we use but their internal geometric structure.

We know that Ω is the least uncountable ordinal. Up to now we have use only ordinals below Ω , in the stretch

$$\text{Ord} = [0, \Omega)$$

and, in fact, only a small part of this stretch. Let

$$\Omega^+ = \Omega^{\Omega^{\Omega^{\dots}}}$$

the uncountable analogue of ϵ_0 . We know that Ω is critical, and Ω^+ is the next critical.

We are going to use ordinals $\xi < \Omega^+$ to describe the shape of a constructed helpful function. We need the material of Section 3 of [Com]. I won't repeat all the details here but I will give enough to indicate what is going on.

The Cantor normal form gives us a description of each countable ordinal below ϵ_0 , and a similar method gives us a description of each ordinal below Ω^+ . For each $\xi < \Omega^+$ we have

$$\xi = \Omega^{\xi(s)} \cdot \alpha(s) + \dots + \Omega^{\xi(0)} \cdot \alpha(0)$$

for a unique strictly descending chain

$$\xi > \xi(s) > \dots > \xi(0)$$

of exponents and associated countable multipliers $\alpha(s), \dots, \alpha(0)$. This is the canonical expansion of ξ .

Sometimes we need not be so tight. Since

$$\Omega^n \cdot \gamma + \Omega^n \cdot \beta = \Omega^n \cdot \alpha$$

there are times where we may use a slack decomposition, one with

$$\xi > \xi(s) \geq \dots \geq \xi(0)$$

with appropriate multipliers. This is particularly useful when we look at the right hand end of the decomposition. We will return to this point when we look at limit ordinals below Ω^+ in more detail.

The canonical decomposition gives us a picture of an ordinal $\xi < \Omega^+$. Each exponent in the decomposition is a smaller ordinal, and this has its own canonical decomposition. The resulting exponents have their own decomposition, and so on. Thus to describe the ordinal $\xi < \Omega^+$ is suffices to give the resulting multipliers in the full decomposition and the position in which they occur. To do this we use a little trick, and to describe that we take a short wander through the woods.

14.2 DEFINITION. A *tree* (or more precisely, a finite rooted tree) is a finite poset with a unique lest element, the **root** and such that for each node a is principal lower section $\downarrow a$ below a is linearly ordered (by the inherited comparison). ■

It is useful to have a more generic view of the way a tree can be grown.
The singleton



is a tree. The unique node is the root of the tree, and its sole leaf.

Suppose

$$\nabla(s), \dots, \nabla(0)$$

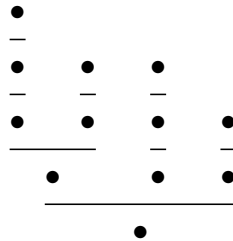
are trees. Then

$$\frac{\nabla(s), \dots, \nabla(0)}{\quad}$$



is a tree. Furthermore, each tree with more than one node is grown in this way from certain smaller trees.

For instance



is a tree with 12 nodes of which 4 are leaves and 1 is the root.

We use certain decorated trees. Consider an arbitrary tree, replace the root by \perp and replace each other node by an ordinal $< \Omega$. Different non-roots can be replaced by different ordinals. For instance

$$\begin{array}{cccc} 0 & & & \\ \hline \alpha & 0 & 0 & \\ \hline \beta & \gamma & \delta & 0 \\ \hline & 1 & \epsilon & \zeta \\ \hline \perp & & & \end{array}$$

is a decorated version of the previous tree, and this will occur again in a later example.

Each ordinal $\xi < \Omega$ gives us a decorated tree. This will have the form

$$\nabla(\xi) = \frac{\begin{array}{c} \vdots \\ \vdots \end{array}}{\perp}$$

where the root is decorated with \perp and every other root is decorated with an ordinal $\beta < \Omega$. Given such a tree $\nabla(\xi)$ and an ordinal $\alpha < \omega$ we write

$$\nabla(\xi)_\alpha$$

for the decorated tree formed from $\nabla(\xi)$ by replacing \perp at the root by α .

With these conventions we can describe how each ordinal $\xi < \Omega^+$ gives a decorated tree.

14.3 DEFINITION. For each ordinal $\xi < \Omega^+$ the decorated tree

$$\nabla(\xi)$$

is generated by recursion over the structure of ξ .

For the base case $\xi = 0$ we set

$$\nabla(0) = \frac{0}{\perp}$$

a 2-node decorated tree.

For a non-zero ordinal ξ let

$$\xi = \Omega^{\xi(s)} \cdot \alpha(s) + \cdots + \Omega^{\xi(0)} \cdot \alpha(0)$$

be its canonical expansion. Then

$$\nabla(\xi) = \frac{\nabla(\xi(s))_{\alpha(s)} \cdots \nabla(\xi(0))_{\alpha(0)}}{\perp}$$

using the modified tree from the exponents. ■

In other words, all we are doing is taking the full canonical expansion of ξ and then setting down in a pictorial form the various occurring ordinals below Ω in a way that enables us to reproduce the original full canonical expansion.

Let's look at an example.

14.4 EXAMPLE. Consider

$$\xi = \Omega^{\Omega^\alpha \cdot \beta + \gamma} + \Omega^\delta \cdot \epsilon + \zeta$$

where $\alpha, \beta, \gamma, \delta, \epsilon, \zeta$ are small ordinals. We must be careful here for what we have just seen is *not* the full canonical expansion of ξ . That is

$$\xi = \Omega^{\xi(2)} \cdot 1 + \Omega^{\xi(1)} \cdot \epsilon + \Omega^{\xi(0)} \cdot \zeta$$

where

$$\xi(0) = 0 \quad \xi(1) = \Omega^0 \cdot \delta \quad \xi(2) = \Omega^\eta \cdot \beta + \Omega^0 \cdot \gamma \quad \eta = \Omega^0 \cdot \alpha$$

are the relevant components. We have

$$\nabla(0) = \frac{0}{\perp}$$

so that

$$\nabla(\eta) = \frac{\nabla(0)_\alpha}{\perp} = \frac{0}{\alpha} \quad \text{with} \quad \nabla(\xi(1)) = \frac{\delta}{\perp}$$

by a similar calculation. Also

$$\nabla(\xi(2)) = \frac{\nabla(\eta)_\beta \quad \nabla(0)_\gamma}{\perp} = \frac{\frac{0}{\alpha} \quad 0}{\beta \quad \gamma} \quad \perp$$

so that

$$\nabla(\xi) = \frac{\nabla(\xi(2))_1 \quad \nabla(\xi(1))_\epsilon \quad \nabla(\xi(0))_\zeta}{\perp}$$

which is the decorated tree that appeared above. ■

The bound $|\xi|$ of an ordinal is the maximum ordinal that occurs in $\nabla(\xi)$. A recursive construction of this is given by Definition 3.1 of [Com].

I think that's enough about the christmas decorations otherwise I might be tempted to bring on the fairy lights.

We come now to the crucial difference between ordinals below Ω and ordinals below Ω^+ . For the longer stretch there are two kinds of limit ordinals, and we must treat these very differently.

The (tight) canonical expansion of a limit ordinal $\xi < \Omega^+$ has one of three forms, namely

$$(1) \quad \xi = \Sigma + \Omega^\Delta \cdot \mu \quad (2) \quad \xi = \Sigma + \Omega^\Delta \cdot (\delta + 1) \quad (3) \quad \xi = \Sigma + \Omega^{\Pi+1} \cdot (\delta + 1)$$

where

$$(1) \quad \mu < \Omega \text{ is a limit ordinal} \quad (2) \quad \Delta < \Omega^+ \text{ is a limit ordinal} \quad (3) \quad \text{no restriction}$$

and δ and Π are arbitrary. The ordinal Σ is the first part of the canonical expansion, so is either 0 or meshes with the displayed end part. We attach a length $\ell(\xi)$ to each such limit ordinal by

$$(1) \quad \ell(\xi) = \mu \quad (2) \quad \ell(\xi) = \ell(\Delta) \quad (3) \quad \ell(\xi) = \Omega$$

respectively. This splits the limit ordinals ξ into two kinds,

$$\text{(Short) Those with } \ell(\xi) < \Omega \quad \text{(Long) Those with } \ell(\xi) = \Omega$$

respectively. We treat these two kinds quite differently. Notice that form (1) is always short, and form (3) is always long. However, form (2) can flit between the two. It is this flitting that causes much of the problems in the development.

For what we do here we don't need all of the last part of the canonical expansion. We can work with the very last bit. Thus we may write

$$(1) \quad \xi = \Gamma + \Omega^\Delta \cdot \mu \quad (2) \quad \xi = \Gamma + \Omega^\Delta \quad (3) \quad \xi = \Gamma + \Omega^{\Pi+1} = \Gamma + \Omega^\Pi \cdot \Omega$$

where

$$(1) \quad \Gamma = \Sigma \quad (2) \quad \Gamma = \Sigma + \Omega^\Delta \cdot \delta \quad (3) \quad \xi = \Sigma + \Omega^{\Pi+1} \cdot \delta$$

respectively. In some way this slack decomposition of ξ gives a better idea of the nature of ξ . It highlights that part which determines the length $\ell(\xi)$.

We attach a limiting sequence

$$(\xi(\alpha) \mid \alpha < \ell(\xi))$$

to each limit ordinal ξ . Thus

$$(1) \quad \xi(\alpha) = \Gamma + \Omega^\Delta \cdot \alpha \quad (2) \quad \xi(\alpha) = \Gamma + \Omega^{\Delta(\alpha)} \quad (3) \quad \xi(\alpha) = \Gamma + \Omega^{\Pi+1} = \Gamma + \Omega^\Pi \cdot \alpha$$

respectively. This is a kind of analogue of the notion of a fundamental sequence for a limit ordinal $\lambda < \Omega$. However, note that this limiting sequence is usually longer than ω . It is routine to check that this is an ascending sequence of ordinals with

$$\xi = \bigvee \{ \xi(\alpha) \mid \alpha < \ell(\xi) \}$$

in each case. Notice that the proof of this for cases (1, 3) is immediate, but that for case (2) requires an induction. This is typical of the nature of the proofs for such limit ordinals.

A proof that each ordinal $\xi < \Omega^+$ has a certain property will proceed by induction. This may be a progressive induction or a

Base-Step-Leap-Vault

induction. The Leap case obtains the result for a sort limit ordinal, and the Vault case obtains the result for a long limit ordinal. Even in a progressive induction we often have to distinguish between the two kinds of limit ordinals.

As an example of this let's look at a generalization of Lemmas 3.3 and 3.5 of [Com].

14.5 LEMMA. *For each limit ordinal $\xi < \Omega^+$ and each limit ordinal $\nu < \ell(\xi)$ we have*

$$\xi(\nu) = \bigvee \{ \xi(\alpha) \mid \alpha < \nu \}$$

and this is a limit ordinal with $\ell(\xi(\nu)) = \nu$ and

$$\xi(\nu)(\alpha) = \xi(\alpha)$$

for each $\alpha < \nu$.

Proof. We proceed by a progressive induction looking at the three cases

$$(1) \quad \xi = \Gamma + \Omega^\Delta \cdot \mu \quad (2) \quad \xi = \Gamma + \Omega^\Delta \quad (3) \quad \xi = \Gamma + \Omega^{\Pi+1}$$

in unison. For these we have

$$(1) \quad \ell(\xi) = \mu \quad (2) \quad \ell(\xi) = \ell(\Delta) \quad (3) \quad \ell(\xi) = \Omega$$

with

$$(1) \quad \xi(\nu) = \Gamma + \Omega^\Delta \cdot \nu \quad (2) \quad \xi(\nu) = \Gamma + \Omega^{\Delta(\nu)} \quad (3) \quad \xi(\nu) = \Gamma + \Omega^\Pi \cdot \nu$$

for each limit ordinal $\nu < \ell(\xi)$.

From these for cases (1, 3) we see that $\xi(\nu)$ is a limit ordinal with $\ell(\xi(\nu)) = \nu$. The other two parts are then immediate.

Case (2) is not so straight forward. We are given that Δ is a limit ordinal with $\ell(\Delta) = \ell(\xi)$. By the induction hypothesis, for each limit ordinal $\nu < \ell(\xi)$ we have that $\Delta(\nu)$ is a limit ordinal with

$$\ell(\Delta(\nu)) = \nu \quad \Delta(\nu)(\alpha) = \Delta(\alpha)$$

for each ordinal $\alpha < \nu$. But now $\xi(\nu)$ is a limit ordinal of type (2) with

$$\ell(\xi(\nu)) = \ell(\Delta(\nu)) = \nu$$

and

$$\xi(\nu)(\alpha) = \Gamma + \Omega^{\Delta(\nu)(\alpha)} = \Gamma + \Omega^{\Delta(\alpha)} = \xi(\alpha)$$

for each ordinal $\alpha < \nu$. Finally, again by the induction hypothesis

$$\Delta(\nu) = \bigvee \{ \Delta(\alpha) \mid \alpha < \nu \}$$

and hence standard ordinal arithmetic gives

$$\bigvee \{ \xi(\alpha) \mid \alpha < \nu \} = \bigvee \{ \Gamma + \Omega^{\Delta(\alpha)} \mid \alpha < \nu \} = \Gamma + \Omega^{\bigvee \{ \Delta(\alpha) \mid \alpha < \nu \}} = \Gamma + \Omega^{\Delta(\nu)} = \xi(\nu)$$

as required. ■

15 The curly gadgets

As explained in Section 14 we think of each ordinal $\xi < \Omega^+$ not as an ordinal in the usual sense but as the code for a finite tree of countable ordinals, a template. The tree is given by the full canonical decomposition of ξ . In fact, if we are willing to use a more liberal version of addition then we can read ‘countable ordinal, as ‘countable iteration’. Of course, even though we are more interested in the ‘shape’ of an ordinal this does not preclude the use of ordinal recursion (for certain constructions) and ordinal induction (for certain proofs). In this section we show how use use this these template facilities to describe certain helpful functions on various levels.

Consider those functions built up from

$$[0], [1], [2], \dots$$

by the allowable constructions of composition, application, and ordinal iteration for ordinals $\alpha < \Omega$. This gives a whole battery of helpful functions

$$\text{Ord}^{(l+1)} \longrightarrow \text{Ord}^{(l+1)}$$

on the various levels. We are going to show that each such function is determined by the shape of a ordinal $\xi < \Omega^+$. Conversely, each such ordinal $\xi < \Omega^+$ determines such a function

$$\{\!\! \{\xi\}\!\!\}_l : \text{Ord}^{(l+1)} \longrightarrow \text{Ord}^{(l+1)}$$

on each level, as indicated.

A similar idea is used in [Com] to produced gadgets

$$\langle \xi \rangle_l : \text{Ord}^{(l)}$$

on all levels. The two families

$$\{\!\! \cdot \!\!\}_\bullet \quad \langle \cdot \rangle_\bullet$$

are not the same but are connected. This relationship is describe at the end of this section.

15.1 DEFINITION. For each level l and ordinal $\xi < \Omega^+$ we generate the helpful function

$$\{\!\! \{\xi\}\!\!\}_l : \text{Ord}^{(l+2)}$$

by

$$\{\!\! \{0\}\!\!\}_l = \mathbf{id}^{(l+2)}$$

for the base, and then

$$\{\!\! \{\xi\}\!\!\}_l = \{\!\! \{\Sigma + \Omega^\Delta \cdot \delta\}\!\!\}_l = (\{\!\! \{\Delta\}\!\!\}_{l+1} [l])^\delta \circ \{\!\! \{\Sigma\}\!\!\}_l$$

where

$$\xi = \Sigma + \Omega^\Delta \cdot \delta$$

is the end part of the canonical decomposition of ξ . ■

More explicitly, if

$$\xi = \Omega^{\xi(s)} \cdot \alpha(s) + \cdots + \Omega^{\xi(0)} \cdot \alpha(0)$$

is the canonical expansion of ξ , then

$$\{\xi\}_l = (\{\xi(0)\}_{l+1}[\iota])^{\alpha(0)} \circ \cdots \circ (\{\xi(s)\}_{l+1}[\iota])^{\alpha(s)}$$

is the helpful function associated with ξ on level l .

Let's look at a few examples.

15.2 EXAMPLES. By definition we have

$$\{0\}_l = \mathbf{id}^{(l+2)}$$

the identity function on $\text{Ord}^{(l+1)}$.

For each countable ordinal α we have

$$\{\alpha\}_l = \{0 + \Omega^0 \cdot \alpha\}_l = (\{0\}_{l+1}[\iota])^\alpha \circ \{0\}_l = [\iota]^\alpha$$

with

$$\{1\}_l = [\iota]$$

as a particular case.

We have

$$\{\Omega\}_l = \{0 + \Omega^1 \cdot 1\}_l = \{1\}_{l+1}[\iota] \circ \{0\}_l = [\iota+1][\iota]$$

and then we find that

$$\{\Omega^\beta \cdot \alpha\}_l = ([\iota+1]^\beta[\iota])^\alpha$$

for each pair α, β of countable ordinals.

We have

$$\{\Omega^\Omega\}_l = \{0 + \Omega^\Omega \cdot 1\}_l = \{\Omega\}_{l+1}[\iota] = [\iota+2][\iota+1][\iota]$$

and then we find that

$$\{\Omega^{\Omega^\gamma \cdot \beta} \cdot \alpha\}_l = \left(([\iota+2]^\gamma[\iota+1])^\beta[\iota] \right)^\alpha$$

for each triple α, β, γ of countable ordinals. ■

By perusing these and similar examples we begin to see how the shape of an ordinal ξ encodes a legal composite of

$$[0], [1], [2], \dots$$

on each level from the second upwards. In fact, every legal compound of these basic helpful functions can be encoded by an ordinal $\xi < \Omega^+$. This is the sole reason that we use the ordinals $\xi < \Omega^+$. We are not interested in their size or length, it is the shape that attracts us.

In Definition 15.1 we assume we deal with ξ via its canonical decomposition. In fact, we need not be so restrictive.

15.3 LEMMA. For each pair of ordinals ξ, χ and level l we have

$$\{\xi + \chi\}_l = \{\chi\}_l \circ \{\xi\}_l$$

provided ξ and χ mesh correctly (that is no part of ξ is lost in the sum $\xi + \chi$).

Proof. We proceed by a progressive induction over χ . To this end let

$$\chi = \Sigma + \Omega^\Delta \cdot \delta$$

be the latter end of the canonical decomposition of χ , and assume that χ does mesh with ξ . This ensure that Σ meshes with ξ . We see that

$$\xi + \chi = (\xi + \Sigma) + \Omega^\Delta \cdot \delta$$

is the latter end of the canonical decomposition of $\xi + \chi$. This gives

$$\{\xi + \chi\}_l = (\{\Delta\}_{l+1}[\iota])^\delta \circ \{\xi + \Sigma\}_l$$

and

$$\{\chi\}_l \circ \{\xi\}_l = (\{\Delta\}_{l+1}[\iota])^\delta \circ \{\Sigma\}_l \circ \{\xi\}_l$$

so an application of the induction hypothesis to Σ gives the required result. ■

By taking χ countable we obtain the following.

15.4 COROLLARY. For each pair of ordinals $\xi < \Omega^+$ and $\alpha < \Omega$ we have

$$\{\xi + \alpha\}_l = [\iota]^\alpha \circ \{\xi\}_l$$

provided the sum $\xi + \alpha$ meshes correctly.

This show us what happens as we step from one ordinal to its successor. However, as we might expect, it is the leap or vault across a limit ordinal that holds the interest.

15.5 LEMMA. Let $\xi < \Omega^+$ be a short limit ordinal. We have

$$\{\xi\}_l = \bigvee \{\{\xi(\alpha)\}_l \mid \alpha < \ell(\xi)\}$$

for each level l (provided each side is supplied only with helpful inputs).

Proof. In more detail we require

$$\{\xi\}_l h = \bigvee \{\{\xi(\alpha)\}_l h \mid \alpha < \ell(\xi)\}$$

for each helpful function h on the appropriate level We proceed by a progressive induction over ξ with allowable variation of l and h .

There are two cases to consider depending on the shape of ξ .

- (1) $\xi = \Sigma + \Omega^\Delta \cdot \mu$
Here $\mu < \Omega$ is a limit ordinal.

- (2) $\xi = \Sigma + \Omega^\Delta \cdot (\delta + 1) = \Gamma + \Omega^\Delta$ where $\Gamma = \Sigma + \Omega^\Delta \cdot \delta$
Here Δ is a short limit ordinal.

We deal with these two cases separately.

(1) We have

$$\xi(\alpha) = \Sigma + \Omega^\Delta \cdot \alpha$$

for each $\alpha < \mu$. We have

$$\{\xi\}_l h = (\{\Delta\}_{l+1} [l])^\mu H \quad \{\xi(\alpha)\}_l h = (\{\Delta\}_{l+1} [l])^\alpha H$$

where

$$H = \{\Sigma\}_l h$$

is the auxiliary helpful function. Since

$$\ell(\xi) = \mu$$

the required result is a standard property of countable iteration.

(2) We have

$$\xi(\alpha) = \Gamma + \Omega^{\Delta(\alpha)}$$

for each $\alpha < \Omega$. We also have

$$\{\xi\}_l h = \{\Delta\}_{l+1} [l] H \quad \{\xi(\alpha)\}_l h = \{\Delta(\alpha)\}_{l+1} [l] H$$

where

$$H = \{\Gamma\}_l h$$

is the auxiliary helpful function. Thus the required result follows by the induction hypothesis applied to Δ . ■

Of course, long limit ordinals do something quite different.

15.6 LEMMA. *Let $\xi < \Omega^+$ be a long limit ordinal. For some level l let*

$$h : \mathbb{H}^{(l+1)}, h_l : \mathbb{H}^{(l)}, \dots, h_1 : \mathbb{H}^{(1)}$$

be a list of helpful functions.

For each ordinal $\zeta < \Omega$ the three ordinals

(a) *the least ν with $\zeta < \nu = \{\xi(\nu)\}_l h h 0$*

(b) *the least ν with $0 < \nu = \{\xi(\nu)\}_l h h \zeta$*

(c) *$\{\xi\}_l h h \zeta$*

are equal.

Proof. We proceed by a progressive induction over ξ with allowable variation of the other gadgets including the level l .

There are two cases to consider depending on the shape of ξ .

$$(2) \quad \xi = \Sigma + \Omega^\Delta \cdot (\delta + 1) = \Gamma + \Omega^\Delta \quad \text{where} \quad \Gamma = \Sigma + \Omega^\Delta \cdot \delta$$

$$(3) \quad \xi = \Sigma + \Omega^{\Pi+1} \cdot (\delta + 1) = \Gamma + \Omega^{\Pi+1} \quad \text{where} \quad \Gamma = \Sigma + \Omega^{\Pi+1} \cdot \delta$$

Here Δ is also a long limit ordinal. We deal with these two cases separately.

(2) We have

$$\xi(\alpha) = \Gamma + \Omega^{\Delta(\alpha)}$$

for each $\alpha < \Omega$. We also have

$$\{\xi\}_l h = \{\Delta\}_{l+1} [l] H \quad \{\xi(\alpha)\}_l h = \{\Delta(\alpha)\}_{l+1} [l] H$$

where

$$H = \{\Gamma\}_l h$$

is the auxiliary helpful function. Thus

$$(a) \quad \text{the least } \nu \text{ with } \zeta < \nu = \{\Delta(\nu)\}_{l+1} [l] H h 0$$

$$(b) \quad \text{the least } \nu \text{ with } 0 < \nu = \{\Delta(\nu)\}_{l+1} [l] H h \zeta$$

$$(c) \quad \{\Delta\}_{l+1} [l] H h \zeta$$

are the three ordinals we are interested in. The induction hypothesis applied to Δ immediately gives the required result.

(3) We have

$$\xi(\alpha) = \Gamma + \Omega^\Pi \cdot \alpha$$

for each $\alpha < \Omega$.

With

$$H = \{\Gamma\}_l h$$

we have

$$\{\xi\}_l h = \{\Pi + 1\}_{l+1} [l] H = [l+1] K H$$

where

$$K = \{\Pi\}_{l+1} [l]$$

is the auxiliary helpful function. This second step follows by unravelling $\{\Pi + 1\}_{l+1}$. By a similar calculation we have

$$\{\xi(\alpha)\}_l h = K^\alpha H$$

for each ordinal $\alpha < \Omega$. Thus

$$(a) \quad \text{the least } \nu \text{ with } \zeta < \nu = K^\nu H h 0$$

$$(b) \quad \text{the least } \nu \text{ with } 0 < \nu = K^\nu H h \zeta$$

$$(c) \quad [l+1] K H h \zeta$$

are the three ordinals we are interested in. These are equal by Lemma 5.3 of [FandH]. ■

Lemmas 15.5 and 15.6 the function $\{\xi\}_l$ for a limit ordinal ξ is determined by the functions $\{\xi(\alpha)\}_l$ for $\alpha < \ell(\xi)$. This connection seems to be quite different depending on whether the limit ordinal is short or long. In fact, this connection does have a certain commonality.

15.7 DEFINITION. Let

$$\Gamma_1, \dots, \Gamma_{m+1}$$

be an arbitrary list of ordinals $\Gamma < \Omega^+$. For each level l let

$$k_{l+r+1} = \{\} \Gamma_{r+1} \}\}_{l+r+1} [l+r]$$

for each $0 \leq r \leq m$. This produces helpful functions

$$k_{l+1} : \text{Ord}^{(l+2)}, \dots, k_{l+r+1} : \text{Ord}^{(l+r+2)}, \dots, k_{l+m+1} : \text{Ord}^{(l+m+2)}$$

determined by the initial list. ■

This construction starts from an arbitrary list of ordinals. We can modify the idea to start from a single limit ordinal. (This is something I hadn't noticed before.)

15.8 THEOREM. *For each limit ordinal $\xi < \Omega^+$ there are certain ordinals*

$$\Gamma_0, \Gamma_1, \dots, \Gamma_{m+1}$$

such that for each level l we have

$$\{\} \xi(\alpha) \}\}_l = k_{l+m+1}^\alpha k_{l+m} \cdots k_{l+1} \circ \{\} \Gamma_0 \}\}_l$$

for each ordinal $\alpha < \ell(\xi)$. (The case $m = 0$ must be read with some care).

Proof. The property says that for each helpful function $h : \text{Ord}^{(l+1)}$ we have

$$\{\} \xi(\alpha) \}\}_l h = k_{l+m+1}^\alpha k_{l+m} \cdots k_{l+1} k_l$$

where

$$k_l = \{\} \Gamma_0 \}\}_l h$$

is the new helpful function of type $\text{Ord}^{(l+1)}$. We verify this by a progressive induction on ξ with allowable variation of the other parameters.

As usual we consider the three possible shapes of the limit ordinal ξ . It turns out that shapes (1, 3) are similar and that $m = 0$ in both cases. Thus we show

$$\{\} \xi(\alpha) \}\}_l h = k_{l+1}^\alpha k_l$$

for these two cases.

(1) We have

$$\xi = \Sigma + \Omega^\Delta \cdot \mu$$

where $\mu < \Omega$ is a limit ordinal. We let

$$\Gamma_0 = \Sigma \quad \Gamma_1 = \Delta$$

so that for each $\alpha < \mu = \ell(\xi)$ we have

$$\{\} \xi(\alpha) \}\}_l h = (\{\} \Gamma_1 \}\}_{l+1} [l])^\alpha (\{\} \Gamma_0 \}\}_l h) = k_{l+1}^\alpha k_l$$

as required.

(3) We have

$$\xi = \Gamma + \Omega^{\Pi+1}$$

with

$$\xi(\alpha) = \Gamma + \Omega^{\Pi} \cdot \alpha$$

for each $\alpha < \Omega = \ell(\xi)$. We set

$$\Gamma_0 = \Gamma \quad \Gamma_1 = \Pi$$

and then the calculation is similar to case (1).

(2) We have

$$\xi = \Gamma + \Omega^{\Delta}$$

is a limit ordinal. For this case we apply the induction hypothesis to Δ .

We set

$$\Gamma_0 = \Gamma$$

and

$$\xi(\alpha) = \Gamma_0 + \Omega^{\Delta(\alpha)}$$

for each $\alpha \leq \ell(\Delta) = \ell(\xi)$. For each helpful $h : \text{Ord}^{(l+1)}$ we have

$$\{\! \{ \xi(\alpha) \} \! \}_l h = \{\! \{ \Delta(\alpha) \} \! \}_{l+1} [l] k_\alpha$$

for the helpful function k_α , as above.

We now apply the induction hypothesis to Δ and take a bit of care with the subscripting.

Attach to Δ there are ordinals

$$\Gamma_1, \dots, \Gamma_{m+2}$$

such that with

$$k_{l+r+2} = \{\! \{ \Gamma_{r+2} \} \! \}_{l+r+2} [l+r+1]$$

we have

$$\{\! \{ \Delta(\alpha) \} \! \}_{l+1} [l] = k_{l+m+2}^\alpha \cdots k_{l+2}$$

for each $\alpha < \ell(\Delta) = \ell(\xi)$. Plugging this into the description of $\{\! \{ \xi(\alpha) \} \! \}_l h$ above gives the required result. \blacksquare

In [Com] I used pointy gadgets

$$\langle \cdot \rangle_l : \text{Ord}^{(l)}$$

which are related to the curly gadgets of this section. Let's look at this relationship.

The following is extracted from Definition 4.2 of [Com] and the remarks just after.

15.9 DEFINITION. For each $\xi < \Omega^+$ the gadget

$$\langle \xi \rangle_l : \text{Ord}^{(l)}$$

is obtained by

$$\langle 0 \rangle_0 = \epsilon_0 \quad \langle 0 \rangle_1 = \mathbf{Next} \quad \langle 0 \rangle_{l+2} = [l]$$

and

$$\xi = \Sigma + \Omega^{\Delta} \cdot \delta \quad \text{gives} \quad \langle \xi \rangle_l = \langle \Delta \rangle_{l+1}^{\delta} \langle \Sigma \rangle_l$$

where the left hand side is the end part of the canonical decomposition of ξ . \blacksquare

The essential difference between $\{\cdot\}_\bullet$ and $\langle\cdot\rangle_\bullet$ is that the pointy gadgets have the ϵ_0 and **Next** built into the construction but the curly gadgets do not. There are times when we may want to view ϵ_0 and **Next** as parameters to the construction and which may be changed under certain circumstances. In this sense the curly gadgets are better.

Given the base pointy gadgets $\langle 0 \rangle_\bullet$, the general pointy gadgets can be generated from the curly gadgets as follows.

15.10 LEMMA. *We have*

$$\langle \xi \rangle_{l+1} = \{\xi\}_l \langle 0 \rangle_{l+1}$$

for each ordinal $\xi < \Omega^+$ and level l .

Proof. We proceed by a progressive induction on ξ with allowable variation of l . Although this is not necessary, let's check the base case $\xi = 0$. We have

$$\{0\}_l \langle 0 \rangle_{l+1} = \mathbf{id}^{(l+2)} \langle 0 \rangle_{l+1} = \langle 0 \rangle_{l+1}$$

as required.

For the induction step suppose

$$\xi = \Sigma + \Omega^\Delta \cdot \delta$$

is the end part of the canonical decomposition of ξ . We have

$$\langle \xi \rangle_{l+1} = \langle \Delta \rangle_{l+2}^\delta \langle \Sigma \rangle_{l+1}$$

by the definition of $\langle\cdot\rangle_\bullet$. We also have

$$\langle \Delta \rangle_{l+2} = \{\Delta\}_{l+1} \langle 0 \rangle_{l+2} = \{\Delta\}_{l+1} [l] \quad \langle \Sigma \rangle_{l+1} = \{\Sigma\}_l \langle 0 \rangle_{l+1}$$

by the induction hypothesis. Thus

$$\begin{aligned} \langle \xi \rangle_{l+1} &= \langle \Delta \rangle_{l+2}^\delta \langle \Sigma \rangle_{l+1} \\ &= \langle \Delta \rangle_{l+2}^\delta (\{\Sigma\}_l \langle 0 \rangle_{l+1}) \\ &= (\langle \Delta \rangle_{l+2}^\delta \circ \{\Sigma\}_l) \langle 0 \rangle_{l+1} \\ &= ((\{\Delta\}_{l+1} [l])^\delta \circ \{\Sigma\}_l) \langle 0 \rangle_{l+1} = \{\xi\}_l \langle 0 \rangle_{l+1} \end{aligned}$$

as required. ■

This shows how to compute $\langle\cdot\rangle_{l+1}$ in terms of $\{\cdot\}_l$. But how do we compute $\langle\cdot\rangle_0$?

15.11 COROLLARY. *For each ordinal $\xi < \Omega^+$ we have*

$$\langle \xi \rangle_0 = (\{\Delta\}_0 \mathbf{Next})^\delta \langle \Sigma \rangle_0$$

where

$$\xi = \Sigma + \Omega^\Delta \cdot \delta$$

is the end part of the canonical decomposition of ξ .

Proof. By the construction of $\langle \cdot \rangle_0$ we have

$$\langle \xi \rangle_0 = \langle \Delta \rangle_1^\delta \langle \Sigma \rangle_0$$

so that Lemma 15.10 applied to Δ gives the required result. \blacksquare

Let's gather these connections together in one place. For each ordinal $\xi < \Omega^+$ we have

$$\begin{aligned} (0) \quad \langle \xi \rangle_0 &= (\{\Delta\}_0 \mathbf{Next})^\delta \langle \Sigma \rangle_0 \\ (1) \quad \langle \xi \rangle_1 &= \{\xi\}_0 \mathbf{Next} \\ (l+1) \quad \langle \xi \rangle_{l+2} &= \{\xi\}_{l+1} [l] \end{aligned}$$

for each level l . For level 0 we use the end part of the canonical decomposition of ξ . When ξ is a successor ordinal we can expand these using Corollary 15.4. When ξ is a limit ordinal we can expand using Theorem 15.8. In this way we can obtain an explicit description of each pointy gadget.

In Section 16 we need to use part of Lemma 4.4(<) of [Com]. The part we need is a consequence of Lemma 15.5. Let's check that.

15.12 LEMMA. *For each short limit ordinal $\xi < \Omega^+$ we have*

$$\langle \xi \rangle_0 = \bigvee \{ \langle \xi(\alpha) \rangle_0 \mid \alpha < \ell(\xi) \}$$

with a similar equality at higher levels.

Proof. We proceed by a progressive induction on ξ . As usual, we need to consider two possible cases. Only one of these cases needs to invoke the induction hypothesis.

The two cases are

$$(1) \quad \xi = \Gamma + \Omega^\Delta \cdot \mu \qquad (2) \quad \xi = \Gamma + \Omega^\Delta$$

where

$$(1) \quad \mu < \Omega \text{ is a limit ordinal} \qquad (2) \quad \Delta \text{ is a short limit ordinal}$$

with

$$(1) \quad \ell(\xi) = \mu \qquad (2) \quad \ell(\xi) = \ell(\Delta)$$

respectively. For case (1) we have extracted the end part of a canonical decomposition of ξ , but for case (2) with extracted the relevant part. For these cases we have

$$(1) \quad \xi(\alpha) = \Gamma + \Omega^\Delta \cdot \alpha \qquad (2) \quad \xi(\alpha) = \Gamma + \Omega^{\Delta(\alpha)}$$

for each $\alpha < \ell(\xi)$.

Using Corollary 15.11 we have

$$(1) \quad \langle \xi \rangle_0 = (\{\Delta\}_0 \mathbf{Next})^\mu \langle \Gamma \rangle_0 \qquad (2) \quad \langle \xi \rangle_0 = (\{\Delta\}_0 \mathbf{Next}) \langle \Gamma \rangle_0$$

with

$$(1) \quad \langle \xi(\alpha) \rangle_0 = (\{\Delta\}_0 \mathbf{Next})^\alpha \langle \Gamma \rangle_0 \qquad (2) \quad \langle \xi(\alpha) \rangle_0 = (\{\Delta(\alpha)\}_0 \mathbf{Next}) \langle \Gamma \rangle_0$$

for each $\alpha < \ell(\xi)$.

From this we see that case (1) is immediate by the definition of ordinal iterates. However, case (2) needs the induction hypothesis and Lemma 15.5. \blacksquare

This is the part of Lemma 4.4(<) of [Com] that we need. The other part follows in a similar way.

16 An enumerating function

In Section 5 of [Com] I use a function

$$\psi : [0, \Omega^+) \longrightarrow [0, \Omega)$$

which enumerates critical ordinals in ascending order with plenty of stagnant phases. In this section we take another look at that function. Except for one or two parts I will not repeat material from [Com], but I will set down some details not given there.

The function of [Com] is based on an arbitrary helpful function \mathbf{N} and an arbitrary starting critical ordinal ϵ . Here we look at the case $\mathbf{N} = \mathbf{Next}$ with $\epsilon = \epsilon_0$.

16.1 DEFINITION. The outputs of the enumerating function

$$\psi : [0, \Omega^+) \longrightarrow [0, \Omega)$$

are generated by recursion on the inputs.

Consider $\xi < \Omega^+$ and suppose $\psi\eta$ is known for all $\eta < \xi$. Then $\psi\xi$ is the least critical ordinal $\theta < \Omega$ with

$$\left. \begin{array}{l} |\eta| < \theta \\ \eta < \xi \end{array} \right\} \implies \psi\eta < \theta$$

for all $\eta < \Omega^+$. ■

By construction this function ψ is defined on an initial stretch of $[0, \Omega^+)$ and outputs only criticals. In the first part of this section we show that it is defined on the whole of $[0, \Omega^+)$. Almost trivially this function ψ is monotone, and this is Lemma 5.2(a) of [Com]. A few moment's thought shows that it doesn't miss out any criticals, and this is Lemma 5.4 of [Com]. Accepting this, on cardinality grounds the function ψ must be constant for long periods.

16.2 EXAMPLE. Vacuously we have $\psi 0 = \epsilon_0$ (for there are no $\eta < 0$).

Let ν be the least critical with $\nu = \epsilon_\nu$. We show

$$\alpha < \nu \implies \psi\alpha = \epsilon_\alpha$$

by a progressive induction on α . Consider any such α . For each $\beta < \alpha$ the induction hypothesis gives

$$\psi\beta = \epsilon_\beta < \epsilon_\alpha$$

so that $\psi\alpha \leq \epsilon_\alpha$. If $\psi\alpha < \epsilon_\alpha$ then $\psi\alpha = \epsilon_\beta$ for some $\beta < \alpha$. Since $\beta < \nu$ we have

$$\beta < \epsilon_\beta \quad \beta < \alpha$$

and hence $\epsilon_\beta = \psi\alpha < \epsilon_\beta$, which is a contradiction.

A slight modification of this argument shows that $\phi\nu = \nu = \epsilon_\nu$.

What happens beyond ν ? We may check that

$$\nu \leq \alpha < \Omega \implies \psi\alpha = \nu$$

again by a progressive induction on α .

Now look at $\xi = \Omega$. We require a critical θ such that

$$\left. \begin{array}{l} \alpha < \theta \\ \alpha < \Omega \end{array} \right\} \implies \psi\eta < \theta$$

for all countable ordinals α . Since $\psi\alpha \leq \nu = \epsilon_\nu$ we see that

$$\psi\Omega = \epsilon_{\nu+1}$$

and then ψ begins to steadily increase for a while.

Each ordinal $\Omega \leq \xi < \Omega \cdot 2$ has the form $\Omega + \alpha$ for some $\alpha < \Omega$. Furthermore

$$|\Omega + \alpha| = \alpha$$

for such α (unless $\alpha = 0$). Using this we may check that

$$\psi(\Omega + \alpha) = \epsilon_{\nu+\alpha}$$

for all $\alpha \leq \mu$ where μ is the next ordinal with $\mu = \epsilon_\mu$. Then $\psi(\Omega \cdot 2) = \mu$.

In fact for all sufficiently small α we have

$$\psi(\Omega \cdot \alpha) = \nu_\alpha$$

where ν_α enumerates the fixed points of $\beta \mapsto \nu_\beta$.

It is an interesting exercise to see just how far you can go with these calculations without calling for the end of prohibition. ■

The construction of ψ makes use of the bound $|\cdot|$ on the input ordinals. To help with the analysis it is convenient to look at that with a different notation.

16.3 DEFINITION. For each critical ordinal $\theta < \Omega$ let $\Xi(\theta)$ be the subset of $[0, \Omega^+)$ given by

$$\xi \in \Xi(\theta) \iff |\xi| < \theta$$

for $\xi < \Omega^+$. ■

It is interesting to try to get an idea of what $\Xi(\theta)$ looks like. Once again we come across the iterated nesting problem.

16.4 EXAMPLE. Let $\theta < \omega$ be an arbitrary critical. We see that the interval

$$[0, \theta)$$

is an initial part of $\Xi(\theta)$. After that the next ordinal in $\Xi(\theta)$ is Ω , and then

$$[\Omega, \Omega + \theta)$$

is a block in $\Xi(\theta)$. In fact, for a while $\Xi(\theta)$ is built up of blocks

$$[\Omega \cdot \alpha, \Omega \cdot \alpha + \theta)$$

for $\alpha < \theta$. This gives us a certain pattern of blocks.

After this $\Xi(\theta)$ misses out all the next few ordinals and jumps straight to Ω^2 . It then repeats the previous pattern with blocks

$$[\Omega^2 + \Omega \cdot \alpha, \Omega^2 + \Omega \cdot \alpha + \theta)$$

for $\alpha < \theta$.

In fact, this pattern is repeated in the form of blocks starting with

$$\Omega^\gamma \cdot \beta$$

for $\beta, \gamma < \theta$. After that $\Xi(\theta)$ jumps straight to Ω^Ω .

And so on. ■

At first sight it seems that the set $\Xi(\theta)$ could be large. But that is a mistaken view. The following is the crucial observation.

16.5 LEMMA. *For each critical ordinal $\theta < \Omega$ the set $\Xi(\theta)$ is countable.*

Proof. We have

$$\xi \in \Xi(\theta) \iff |\xi| < \theta \iff \nabla(\xi) \subseteq [0, \theta)$$

where $\nabla(\xi)$ is a certain finite tree decorated with countable ordinals. The right hand inclusion means that these decorating ordinals are all in $[0, \theta)$. The assignment

$$\xi \longmapsto \nabla(\xi)$$

is injective.

There are only countably many ordinals $\alpha < \theta$. Thus each finite tree can be decorated with such ordinals in just countably many ways.

There are only countably many finite trees.

Thus there are only countably many tree decorated with ordinals $\alpha < \theta$. Each such tree puts at most one ordinal into $\Xi(\theta)$, and hence this set is countable. ■

This is the crucial result that ensure that ψ is total. In fact, there is a more general result.

16.6 LEMMA. *Consider any $\xi < \Omega^+$ and suppose*

$$\phi : [0, \xi) \longrightarrow [0, \Omega)$$

is any function. Then there is at least one critical ordinal $\theta < \Omega$ such that

$$\left. \begin{array}{l} |\eta| < \theta \\ \eta < \xi \end{array} \right\} \implies \varphi\eta < \theta$$

for all $\eta < \Omega^+$.

Proof. Since $|\xi| < \Omega$, we have $|\xi| < \pi$ for a final section of critical ordinals $\pi < \Omega$. We set up an inflationary function

$$\pi \longmapsto \pi'$$

on these critical ordinals.

Given such a critical π , Lemma 16.5 ensures that the set

$$\Pi = \{\pi\} \cup \{\phi\eta \mid \eta \in \Xi(\pi), \eta < \xi\}$$

is countable. Thus

$$\bigvee \Pi < \Omega$$

and hence

$$\pi < \bigvee \Pi < \pi' < \Omega$$

for some critical ordinal π' . Observe that

$$\left. \begin{array}{l} |\eta| < \pi \\ \eta < \xi \end{array} \right\} \implies \varphi\eta \in \pi \implies \varphi\eta < \pi'$$

for $\eta < \Omega^+$.

Starting with any critical π with $|\xi| < \pi$, we iterate this construction

$$\pi_0 = \pi \quad \pi'_{r+1}$$

to produce an ascending chain

$$\pi = \pi_0 \leq \pi_1 \leq \pi_2 \leq \cdots \leq \pi_r \leq \cdots \quad (r < \omega)$$

of criticals. The supremum

$$\theta = \bigvee \{\pi_r \mid r < \omega\}$$

is critical, and $\theta < \Omega$ on cardinality grounds.

For each $\eta < \Omega^+$ we have

$$\left. \begin{array}{l} |\eta| < \theta \\ \eta < \xi \end{array} \right\} \implies (\exists r) \left[\begin{array}{l} |\eta| \in \pi_r \\ \eta < \xi \end{array} \right] \implies (\exists r)[\varphi\eta < \pi_{r+1}] \implies \varphi\eta < \theta$$

for the required result. ■

This gives us our first main result.

16.7 THEOREM. *The enumerating function ψ is total.*

Proof. By way of contradiction suppose ψ is not total. There is at least one $\xi < \Omega^+$ where $\psi\xi$ is not defined. We look at the smallest such ξ .

We certainly have a function

$$\psi : [0, \xi) \longrightarrow [0, \Omega)$$

(by the minimality of ξ). Thus Lemma 16.6 gives at least one critical θ with

$$\left. \begin{array}{l} |\eta| < \theta \\ \eta < \xi \end{array} \right\} \implies \psi\eta < \theta$$

(for $\eta < \Omega^+$). Taking the smallest such θ we see that $\psi\xi$ is defined, which is the contradiction. ■

With this we can begin to develop the properties of this function. As in Lemma 5.2 of [Com] the function is monotone with a certain continuity property. The outputs of ψ , which are all critical, steadily increase with certain stagnant stretches.

16.8 DEFINITION. An ordinal $\xi < \Omega^+$ is **tame** if $|\xi| < \psi\xi$, and **wild** if $\psi\xi \leq |\xi|$. ■

This is just Definition 5.3 of [Com], but note there is a typo in that version. We come now to a crucial result, namely Lemma 5.4 of [Com].

16.9 LEMMA. For each $\xi < \Omega^+$ we have

$$\psi(\xi + 1) = \begin{cases} \mathbf{Next}\theta & \text{if } \xi \text{ is tame} \\ \theta & \text{if } \xi \text{ is wild} \end{cases}$$

where $\theta = \psi\xi$.

This with the continuity property of Lemma 5.2(b) of [Com] gives us a complete description of ψ . Or rather it would do if we knew when ψ is tame and when it is wild.

To conclude this section we relate ψ to the pointy gadgets $\langle \cdot \rangle_\bullet$ of Section 15. This is just Theorem 6.1 of [Com], but we will give the non-trivial parts of the proof.

16.10 THEOREM. We have

$$\psi\xi = \langle \xi \rangle_0$$

for each tame ordinal $\xi < \Omega^+$.

Proof. This is obtained by a progressive induction on ξ . The base and step cases are immediate. Here we look at the jump across a short limit ordinal and the the vault across a long limit ordinal.

Let ξ be a short limit ordinal. By Lemma 5.2(b) of [Com] and Lemma 15.12 (of Section 15) we have

$$\begin{aligned} \psi\xi &= \bigvee \{ \psi(\xi(\alpha)) \mid \alpha < \ell(\xi) \} \\ \langle \xi \rangle_0 &= \bigvee \{ \langle \xi(\alpha) \rangle_0 \mid \alpha < \ell(\xi) \} \end{aligned}$$

so the induction hypothesis gives the required result.

Let ξ be a long limit ordinal. We invoke various results from [Com].

Let

$$\theta = \psi\xi \quad \pi = \langle \xi \rangle_0$$

so that $\theta = \pi$ is required. By Corollary 5.9 and Lemma 4.4(=) of [Com] we have

$$\begin{aligned} \theta &\text{ is the least ordinal } \nu \text{ with } \nu = \psi(\xi(\nu)) \\ \pi &\text{ is the least ordinal } \nu \text{ with } \nu = \langle \xi(\nu) \rangle_0 \end{aligned}$$

respectively.

Using θ we invoke Lemma 5.8. Thus

$$\begin{aligned} \theta < \alpha &\implies \xi(\alpha) \text{ is wild} \\ \alpha < \theta &\implies \xi(\alpha) \text{ is tame} \end{aligned}$$

for each ordinal $\alpha < \Omega$. In particular, the induction hypothesis gives

$$\psi(\xi(\alpha)) = \langle \xi(\alpha) \rangle_0$$

for each $\alpha < \theta$.

The ordinal

$$\eta = \xi(\theta)$$

is wild, but

$$\psi\eta = \theta$$

by the second part of Lemma 5.8. By Lemma 3.5 we have

$$\ell(\eta) = \theta$$

with

$$\eta(\alpha) = \xi(\alpha)$$

for each $\alpha < \theta$. Thus

$$\psi(\eta(\alpha)) = \psi(\xi(\alpha)) = \langle \xi(\alpha) \rangle_0 = \langle \eta(\alpha) \rangle_0$$

for each $\alpha < \theta$.

By Lemmas 5.2(b) and 4.4(<) we have

$$\begin{aligned} \psi\eta &= \bigvee \{ \psi(\eta(\alpha)) \mid \alpha < \theta \} \\ \langle \eta \rangle_0 &= \bigvee \{ \langle \eta(\alpha) \rangle_0 \mid \alpha < \theta \} \end{aligned}$$

and hence

$$\theta = \psi\eta = \langle \eta \rangle_0 = \langle \xi(\theta) \rangle_0$$

to show that $\pi \leq \theta$.

By way of contradiction suppose that $\pi < \theta$. Then $\xi(\pi)$ is tame and we have

$$\psi(\xi(\pi)) = \langle \xi(\pi) \rangle_0 = \pi$$

by the induction hypothesis. But now a use of Lemma 3.4(d) gives

$$\pi \leq |\xi(\pi)| < \psi(\xi(\pi)) = \pi$$

which is the contradiction. ■

17 Present Development

The enumerating function ψ of Section 16 is based on the enumerating functions currently used to generate ordinals. The use of larger ordinals as templates goes back to Bachmann [4], but idea of an enumerating function as currently used was introduced by Buchholz in [6] with a unpublished manuscript being written a few years earlier. A description of idea can be found in [18, 19, 20], and a more general method is describe in [7]. However, these accounts are not easy to read.

In this section I will describe a standard variant of the current enumerating function, and then in due course I will show it is the same as the enumeration function of Section 16.

Because the two functions are the same we also use ‘ ψ ’ for the currently preferred description. When we do have to compare the two we will be more careful with the notation.

17.1 DEFINITION. The outputs of the enumerating function

$$\psi : [0, \Omega^+) \longrightarrow [0, \Omega)$$

are generated by recursion on the inputs.

Consider $\xi < \Omega^+$ and suppose $\psi\eta$ is known for all $\eta < \xi$. Consider those sets

$$\Xi \subseteq [0, \Omega^+)$$

where

$$(E1) \quad 0, \Omega \in \Xi \quad (E2) \quad \Xi \text{ is closed under } +, \omega^\bullet \quad (E3) \quad \left. \begin{array}{l} \eta \in \Xi \\ \eta < \xi \end{array} \right\} \implies \psi\eta \in \Xi$$

(for $\eta < \Omega^+$). Let $\Psi\xi$ be the least such set Ξ . Then $\psi\xi$ is the least non-member of $\Psi\xi$. ■

This is not exactly the the clearest definition you have ever seen, is it? We are going to do some work on it to sort out what is going on. Let's begin by looking at what the definition actually says.

Suppose $\psi\eta$ is known for all $\eta < \xi$, and we wish to calculate $\psi\xi$. As set up this evaluation is a search for the set $\Psi\xi$. We look at those sets Ξ satisfying $(E1, E2, E3)$. The whole stretch $[0, \Omega^+)$ is one such set. The intersection of any family of such sets is itself such a set. Thus there is a smallest such set, and this is $\Psi\xi$. We require

$$[0, \Omega) \not\subseteq \Psi\xi$$

so we may take $\psi\xi$ to be the smallest ordinal not captured by $\Psi\xi$.

At this stage it is not obvious that ψ is total, that it is defined for all inputs $\xi < \Omega^+$. We will verify this in due course, but for the time being let us say an ordinal $\xi < \Omega^+$ is *acceptable* if $\psi\eta$ exists for all $0 \leq \eta \leq \xi$.

Notice that 0 is acceptable. For the case $\xi = 0$ the condition $(E3)$ is vacuous. So all we need is the smallest set satisfying $(E1, E2)$. A few moment's thought gives $\psi 0 = \epsilon_0$. After that we find that $\psi 1 = \epsilon_1, \psi 2 = \epsilon_2, \dots$, but this behaviour does not continue for ever.

We will eventually show that each $\xi < \Omega^+$ is acceptable. We also develop a better understanding of the set $\Psi\xi$ so that we obtain a neater description of this enumerating function which doesn't even mention Ψ .

There is one obvious requirement which we can get out of the way immediately.

17.2 LEMMA. *For each acceptable $\xi < \Omega^+$ the value $\psi\xi$ is critical.*

Proof. Using the closure property $(E2)$ we have

$$\alpha, \beta \in \Psi\xi \implies \omega^\alpha \cdot m + \beta \in \Psi\xi$$

for all $\alpha, \beta < \Omega$ and $m \in \mathbb{N}$.

Let $\theta = \psi\xi$, so that $[0, \theta) \subseteq \Psi\xi$ with $\theta \notin \Psi\xi$. In the usual way we have

$$\theta = \omega^\alpha \cdot m + \beta$$

for some

$$\alpha \leq \omega^\alpha \leq \theta < \omega^{\alpha+1} \quad \beta < \omega^\alpha \leq \theta$$

and some $m \in \mathbb{N}$. If $\alpha < \theta$ then

$$\theta = \omega^\alpha \cdot m + \beta \in \Psi\xi$$

which is not so. Thus $\alpha = \theta$ to give

$$\theta \leq \omega^\theta \leq \theta$$

and hence θ is critical. ■

This shows that ψ does indeed produce a collection of criticals. But which ones and how does it do it? The problem with Definition 17.1 is the use of the auxiliary set $\Psi\xi$. Let's try to get a better understanding of this set.

17.3 DEFINITION. Let $\theta < \Omega$ be a critical ordinal. A set

$$\Xi \subseteq [0, \Omega^+)$$

is, respectively,

$$\theta\text{-striated}^\omega \quad \theta\text{-striated}^+ \quad \theta\text{-striated}^\Omega$$

if

$$\begin{array}{lll} (S^{\Omega 1}) \quad 0, \Omega \in \Xi & (S^{\Omega 2}) \quad \Xi \text{ is closed under } + & (S^{\Omega 3}) \quad \left. \begin{array}{l} \eta \in \Xi \\ \alpha < \theta \end{array} \right\} \implies \Omega^\eta \cdot \alpha \in \Xi \\ (S^{+1}) \quad 0, \Omega \in \Xi & (S^{+2}) \quad \Xi \text{ is closed under } +, \omega^\bullet & (S^{+3}) \quad \left. \begin{array}{l} \eta \in \Xi \\ \alpha < \theta \end{array} \right\} \implies \eta + \alpha \in \Xi \\ (S^{\omega 1}) \quad 0, \Omega \in \Xi & (S^{\omega 2}) \quad \Xi \text{ is closed under } + & (S^{\omega 3}) \quad \left. \begin{array}{l} \eta \in \Xi \\ \alpha < \theta \end{array} \right\} \implies \omega^\eta \cdot \alpha \in \Xi \end{array}$$

for all $\eta < \omega^+$ and $\alpha < \Omega$.

Let

$$\Xi^\omega(\theta) \quad \Xi^+(\theta) \quad \Xi^\Omega(\theta)$$

be the least such set. ■

In due course we will show that

$$\Xi^\omega(\theta) = \Xi^+(\theta) = \Xi^\Omega(\theta) = \Xi(\theta)$$

where $\Xi(\theta)$ is the set introduced in Section 16. Here we are mainly concerned with $\Xi^+(\theta)$, but by looking at the other three sets we do get a better understanding.

Trivially, the whole stretch $[0, \Omega^+)$ is θ -striated $^\bullet$ of any variety. Also, the intersection of any family of θ -striated $^\bullet$ sets is itself θ -striated $^\bullet$. Thus $\Xi^\bullet(\theta)$, the least θ -striated $^\bullet$ set, does exist.

What does $\Xi^\bullet(\theta)$ look like? By (S1) we have $0 \in \Xi^\bullet(\theta)$, and hence

$$[0, \theta) \subseteq \Xi(\theta)$$

by (S3). A few moment's thought shows that

$$[0, \theta) \cup [\Omega, \Omega^+)$$

is θ -striated $^\bullet$, so this set is an upper bound for $\Xi^\bullet(\theta)$. In particular, we have

$$\alpha \in \Xi^\bullet(\theta) \iff \alpha < \theta$$

for $\alpha < \Omega$. Later we will get a better picture of the uncountable part of $\Xi^\bullet(\theta)$.

Here is why we are interested in θ -striation.

17.4 LEMMA. *Suppose $\xi < \Omega^+$ is acceptable. Then*

$$\Psi\xi = \Xi^+(\psi\xi)$$

and the value $\psi\xi$ is the least critical $\theta < \Omega$ such that

$$\langle \theta \rangle \quad \left. \begin{array}{l} \eta \in \Xi^+(\theta) \\ \eta < \theta \end{array} \right\} \implies \psi\eta < \theta$$

for $\eta < \Omega^+$.

Proof. We show first that $\Psi\xi$ is θ -striated $^+$. To do that only the closure property (S $^+$ 3) is a problem.

Since $[0, \theta) \subseteq \Psi\xi$ we have

$$\left. \begin{array}{l} \eta \in \Psi\xi \\ \alpha < \theta \end{array} \right\} \implies \left\{ \begin{array}{l} \eta \in \Psi\xi \\ \alpha \in \Psi\xi \end{array} \right\} \implies \eta + \alpha \in \Psi\xi$$

to verify (S $^+$ 3). The second implication holds by (E2).

By the minimality of $\Xi(\theta)$, this gives

$$\Xi^+(\theta) \subseteq \Psi\xi$$

so we now need the converse inclusion. To obtain that we verify that the set $\Xi^+(\theta)$ has (E1, E2, E3). As with the first inclusion, only (E3) is a problem.

A use of (S $^+$ 3) with $\eta = 0$ gives $[0, \theta) \subseteq \Xi^+(\theta)$. Thus for arbitrary $\eta < \Omega^+$ the first inclusion gives

$$\left. \begin{array}{l} \eta \in \Xi^+(\theta) \\ \eta < \xi \end{array} \right\} \implies \left\{ \begin{array}{l} \eta \in \Psi\xi \\ \eta < \xi \end{array} \right\} \implies \psi\eta \in \Psi\xi \implies \psi\eta < \theta \implies \psi\eta \in \Xi^+(\theta)$$

for the required result.

This also shows that $\theta = \psi\xi$ satisfies the implication $\langle \theta \rangle$.

Finally, consider any critical θ that satisfies $\langle \theta \rangle$. Then $\Xi^+(\theta)$ satisfies (E1, E2, E3), so

$$\Psi\xi \subseteq \Xi^+(\theta)$$

by the minimality of $\Psi\xi$. If $\theta < \psi\xi$ then $\theta \in \Psi\xi$, so that $\theta \in \Xi^+(\theta)$, which is not so. Thus $\psi\xi \leq \theta$. ■

That's a bit better, isn't it. The evaluation of $\psi\xi$ is a search for a critical θ , the least one such that $\langle \theta \rangle$ holds. The problem with this is that we are not quite sure what $\Xi^+(\theta)$ looks like. And why is the uncountable part of $\Xi^+(\theta)$ needed? Also, we still don't know that ψ is total.

The hypothesis of Lemma 17.4 requires that ξ is acceptable. In fact, another look at the proof gives the following.

17.5 SCHOLIUM. Suppose $\xi < \Omega^+$ and suppose ψ_η is defined for all $\eta < \xi$. Suppose also there is a critical ordinal $\theta < \Omega$ such that $\langle \theta \rangle$ holds. Then $\Psi\xi$ is defined (and $\psi\xi \leq \theta$).

Proof. The condition $\langle \theta \rangle$ ensures that $\Xi^+(\theta)$ is θ -striated⁺. ■

For the next few pages we can forget about ψ and Ψ . We analyse the various notion of striation and the sets $\Xi^\bullet(\theta)$.

We look first at the set $\Xi^\Omega(\theta)$ and find that this is much easier to understand than the official definition suggests.

Recall that from Section 16 for each critical θ we have a set $\Xi(\theta)$ given by

$$\eta \in \Xi(\theta) \iff |\eta| < \theta$$

for each $\eta < \Omega^+$.

17.6 LEMMA. For each critical ordinal $\theta < \Omega$ we have $\Xi^\omega(\theta) = \Xi(\theta)$, that is

$$\eta \in \Xi^\Omega(\theta) \iff |\eta| < \theta$$

for each $\eta < \Omega^+$.

Proof. Trivially, $\Xi(\theta)$ has $(S^{\Omega 1})$ and $(S^{\Omega 2})$, and a few moments thought shows that it also has $(S^{\Omega 3})$. Thus $\Xi(\theta)$ is θ -striated ^{Ω} .

The minimality of $\Xi^\Omega(\theta)$ gives

$$\Xi^\Omega(\theta) \subseteq \Xi(\theta)$$

and so we require the converse inclusion.

Consider any $\eta \in \Xi(\theta)$ and look at its full Ω -decomposition. The set $\Xi^\Omega(\theta)$ -striated ^{Ω} , so by repeated use of $(S^{\Omega 1}, S^{\Omega 2}, S^{\Omega 3})$ we see that $\eta \in \Xi^\Omega(\theta)$. ■

I think that with this description we can agree that this set $\Xi^\Omega(\theta)$ is quite easy to understand. Furthermore, the striation description isn't entirely helpful.

17.7 LEMMA. For each critical ordinal $\theta < \Omega$, the set $\Xi^\Omega(\theta)$ is θ -striated⁺.

Proof. The only problem is to show that $\Xi^\Omega(\theta)$ is closed under ω^\bullet .

Consider any $\eta \in \Xi^\Omega(\theta)$.

If $\eta < \omega$ then $\eta < \theta$, so that $\omega^\eta < \theta$ (since θ is critical), to give $\omega^\eta \in \Xi^\Omega(\theta)$.

Suppose $\Omega \leq \eta$. Then

$$\eta = \Omega^{\eta(0)} \cdot \alpha(0) + \dots + \Omega^{\eta(s)} \cdot \alpha(s) + \alpha$$

for non-zero exponents $\eta(0), \dots, \eta(s) \in \Xi^\Omega(\theta)$, and multipliers $\alpha(0), \dots, \alpha(s), \alpha < \theta$. Let

$$\eta(i) = 1 + \zeta(i)$$

for each $0 \leq i \leq s$. We have $\eta(i) \neq \zeta(i)$ only when $\eta(i) < \omega$. In particular, each $\zeta(i)$ is in $\Xi^\Omega(\theta)$, and hence

$$\zeta = \Omega^{\zeta(0)} \cdot \alpha(0) + \dots + \Omega^{\zeta(s)} \cdot \alpha(s)$$

is also in $\Xi^\Omega(\theta)$.

With this we have

$$\eta = \Omega \cdot (\Omega^{\zeta(0)} \cdot \alpha(0) + \cdots + \Omega^{\zeta(s)} \cdot \alpha(s)) + \alpha = \Omega \cdot \zeta + \alpha$$

and hence

$$\omega^\eta = \omega^{\Omega \cdot \zeta + \alpha} = \omega^{\Omega \cdot \zeta} \cdot \omega^\alpha = (\omega^\Omega)^\zeta \cdot \omega^\alpha = \Omega^\zeta \cdot \beta$$

where $\beta = \omega^\alpha < \theta$.

Since $\Xi^\Omega(\theta)$ is θ -striated $^\Omega$, a use of $(S^\Omega 3)$ now gives $\omega^\eta \in \Xi^\Omega(\theta)$, as required. \blacksquare

This result gives

$$\Xi^+(\theta) \subseteq \Xi^\Omega(\theta)$$

and we now work towards the converse inclusion. To do that we use the notion of θ -striation $^\omega$. In fact, this is the only reason for introducing that idea. In itself it seems to have no intrinsic interest.

17.8 LEMMA. *Let $\theta < \Omega$ be a critical ordinal. Then each θ -striated $^+$ set is also θ -striated $^\omega$.*

Proof. Let Ξ be any θ -striated $^+$ set. We must show that Ξ has $(S^\omega 3)$.

By $(S^+ 2)$ we have

$$\eta \in \Xi \implies \omega^\eta \cdot m \in \Xi$$

for each $\eta < \Omega^+$ and $m \in \Xi$. We use this to show

$$\alpha < \theta \implies (\forall \eta < \Omega^+) [\eta \in \Xi \implies \omega^\eta \cdot \alpha \in \Xi]$$

by a progressive induction on α .

In the usual way, given $\alpha < \theta$ we have

$$\alpha = \omega^\beta \cdot m + \gamma$$

for some

$$\beta \leq \omega^\beta \leq \alpha < \omega^{\beta+1} \quad \gamma < \omega^\beta \leq \alpha$$

and some $m \in \mathbb{N}$. Assuming $\eta \in \Xi$ we have

$$\omega^{\eta+\beta} \cdot m \in \Xi \quad \omega^\eta \cdot \gamma \in \Xi$$

by a combination of $(S^+ 2, S^+ 3)$ for the left hand condition and the induction hypothesis for the right hand condition. But now

$$\omega^\eta \cdot \alpha = \omega^\eta \cdot (\omega^\beta \cdot m + \gamma) = \omega^{\eta+\beta} \cdot m + \omega^\eta \cdot \gamma$$

so that $\omega^\eta \cdot \alpha \in \Xi$ by another use of $(S^+ 2)$. \blacksquare

We now have

$$\Xi^\omega(\theta) \subseteq \Xi^+(\theta) \subseteq \Xi^\Omega(\theta) = \Xi(\theta)$$

for each critical ordinal θ . We now show that

$$\Xi(\theta) \subseteq \Xi^\omega(\theta)$$

and hence show that all four sets are the same.

We need a couple of preliminary lemmas.

17.9 LEMMA. *Let $\theta < \Omega$ be critical. Then*

$$|\xi| < \theta \implies \Omega \cdot \xi \in \Xi^\omega(\theta)$$

for each $\xi < \Omega^+$.

Proof. For convenience let $\Xi = \Xi^\omega(\theta)$.

Recall that

$$\omega^\Omega = \Omega$$

and $\Omega \in \Xi^\omega(\theta)$.

Observe that if $\xi < \Omega$ then $(S^\omega 3)$ gives

$$|\xi| < \theta \implies \xi < \theta \implies \Omega \cdot \xi = \omega^\Omega \cdot \xi \in \Xi$$

so there is no problem with countable ξ .

We now proceed by a progressive induction on ξ making use of this preliminary observation.

Consider any $\xi < \Omega^+$ with $|\xi| < \theta$. By the observation above we may suppose that $\Omega \leq \xi$. There is a canonical decomposition

$$\xi = \Omega^{\xi(s)} \cdot \alpha(s) + \cdots + \Omega^{\xi(1)} \cdot \alpha(1) + \alpha(0)$$

where

$$\xi(i) < \xi \quad |\xi(i)| \leq |\xi| < \theta \quad \alpha(i) < \theta$$

for each index $1 \leq i \leq s$, and with $\alpha(0) < \theta$. We have

$$\Omega \cdot \xi = \Omega^{1+\xi(s)} \cdot \alpha(s) + \cdots + \Omega^{1+\xi(1)} \cdot \alpha(1) + \Omega \cdot \alpha(0)$$

so by $(S^\omega 2)$ it suffices to show that

$$\Omega^{1+\xi(i)} \cdot \alpha(i) \in \Xi \quad \Omega \cdot \alpha(0) \in \Xi$$

for each index i . The right hand case is dealt with above. Thus it suffices to show that

$$\Omega^{1+\eta} \cdot \beta \in \Xi$$

for all $\eta < \xi$ with $|\eta| < \theta$ and all $\beta < \theta$.

Consider such a pair η and β . We first check that $1 + \eta < \xi$.

If $\eta < \omega$ then

$$1 + \eta < \Omega \leq \xi$$

and if $\omega \leq \eta$ then

$$1 + \eta = \eta < \xi$$

as required.

We now have

$$\Omega \cdot (1 + \eta) \in \Xi$$

by the induction hypothesis. Since $\beta < \theta$ this gives

$$\omega^{\Omega \cdot (1+\eta)} \cdot \beta \in \Xi$$

by ($S^\omega 3$). But

$$\omega^{\Omega \cdot (1+\eta)} = (\omega^\Omega)^{1+\eta} = \Omega^{1+\eta}$$

so that

$$\Omega^{1+\eta} \cdot \beta = \omega^{\Omega \cdot (1+\eta)} \cdot \beta \in \Xi$$

as required. ■

17.10 COROLLARY. *Let $\theta < \Omega$ be critical. Then*

$$\left. \begin{array}{l} |\xi| < \theta \\ \alpha < \theta \end{array} \right\} \implies \Omega^\xi \cdot \alpha \in \Xi^\omega(\theta)$$

for each $\xi < \Omega^+$ and $\alpha < \Omega$.

Proof. Consider any such pair ξ, α . We have

$$\Omega \cdot \xi \in \Xi^\omega(\theta)$$

by Lemma 17.9, and hence

$$\omega^{\Omega \cdot \xi} \cdot \alpha \in \Xi^\omega(\theta)$$

by ($S^\omega 3$). Since

$$\omega^{\Omega \cdot \xi} = (\omega^\Omega)^\xi = \Omega^\xi$$

this gives the required result. ■

With this we have the main result.

17.11 THEOREM. *Let $\theta < \Omega$ be critical. Then*

$$\Xi^\omega(\theta) = \Xi^+(\theta) = \Xi^\Omega(\theta) = \Xi(\theta)$$

holds.

Proof. We have

$$\Xi^\omega(\theta) \subseteq \Xi^+(\theta) \subseteq \Xi^\Omega(\theta) = \Xi(\theta)$$

so it suffices to show

$$\Xi(\theta) \subseteq \Xi^\omega(\theta)$$

that is

$$\xi < |\theta| \implies \xi \in \Xi^\omega(\theta)$$

for each $\xi < \Omega^+$.

Consider any such ξ with $|\xi| < \theta$. We have a canonical decomposition

$$\xi = \Omega^{\xi(s)} \cdot \alpha(s) + \dots + \Omega^{\xi(0)} \cdot \alpha(0)$$

where

$$\xi(i) < \xi \quad |\xi(i)| \leq |\xi| < \theta \quad \alpha(i) < \theta$$

for each index $0 \leq i \leq s$. By Corollary 17.10 we have

$$\Omega^{\xi(i)} \cdot \alpha(i) \in \Xi^\omega(\theta)$$

so that $(S^\omega 2)$ gives the required result. ■

With this result we can take another look at Lemma 17.4. We see that for each $\xi < \Omega^+$ the value $\psi\xi$ is the least critical $\theta < \Omega$ such that

$$\left. \begin{array}{l} \eta \in \Xi(\theta) \\ \eta < \xi \end{array} \right\} \implies \psi\eta < \theta$$

for $\eta < \Omega^+$. This is exactly the construction of the enumerating function ψ of Section 16. Thus we have the following.

17.12 THEOREM. *The two enumerating functions ψ of Section 16 and this section are the same.*

I think we can now forget about Definition 17.1 and stick with the enumerating function of Section 16.

References

- [λH] H. Simmons: An applied λ -calculus for iteration templates.
Available in this part of my web-page
- [Sch] H. Simmons: Generating ordinal notations from below with a non-recursive construction of the Schütte brackets.
Available in this part of my web-page
- [Com] H. Simmons: A comparison of two systems of ordinal notations, *Archive for Math. Logic* 34 (2004) 65-83.
- [FandH] H. Simmons: Fruitful and helpful ordinal functions *Archive for Math. Logic* 47 (2008) 677-709.
- [1] P. Aczel: Describing ordinals using functionals of transfinite type, *J. Symbolic Logic*, 37 (1972) 35 – 47.
- [2] P. Aczel, H. Simmons, S. Wainer: *Proof Theory* Cambridge University Press, 1992.
- [3] J. Avigad and S. Feferman: Gödel’s Functional (“Dialectica”) Interpretation, pp 337 – 405 of [8].
- [4] H. Bachmann: Die Normalfunktionen und das Problem der ausgezeichneten Folgen von Ordnungszahlen, *Vierteljahresschr. Naturforsch. Ges. Zürich* 95 (1950) 115 – 147.
- [5] Francis Borceux: Handbook of categorical algebra, vol. 1, C.U.P (1994).

- [6] W. Buchholtz: A new system of proof-theoretic ordinals, *Ann. Pure and Applied Logic* 32 (1986) 195-207.
- [7] W. Buchholtz and K. Schütte: *Proof Theory of Impredicative Subsystems of Analysis*, Bibliopolis, Naples, 1998.
- [8] S. R. Buss: *Handbook of Proof Theory* North Holland, 1998.
- [9] J. N. Crossley and J. B. Kister: Natural well-orderings, *Arch. math. Logik* 26 (1986.7) 57 – 76.
- [10] N. Danner: Ordinal notations in typed λ -calculi, Ph. D. thesis, Indiana, 1999.
- [11] N. Danner: Ordinals and ordinals functions representable in the simply typed λ -calculus, *Annals of P. and A. Logic* 97 (1999) 179 – 201.
- [12] M. Fairtlough and S. S. Wainer: Hierarchies of provably recursive functions, pp. 149–207 of [8].
- [13] S. Feferman: Hereditarily replete functionals over the ordinals, pp 289 – 301 of [15].
- [14] J-Y. Girard: *Proof Theory and Logical Complexity*, vol. 1, Bibliopolis, 1987.
- [15] A. Kino, J. Myhill, and R. E. Vesley: *Intuitionism and Proof Theory (Buffalo. New York, 1968)*, North-Holland, Amsterdam, 1968.
- [16] J-Y. Girard, Y. Lafont and P. Taylor: *Proofs and Types*, Cambridge University Press, 1996.
- [17] W. A. Howard: A system of abstract constructive ordinals *J. Symbolic Logic* 37 (1972) 355 – 374.
- [18] W. Pohlers: *Proof Theory*, Springer L.N.M 1407, Springer-Verlag, Berlin, 1989.
- [19] W. Pohlers: Proof theory and ordinal analysis, *Arch. Math. Logik* 30 (1991) 311 – 376.
- [20] W. Pohlers: A short course in ordinal analysis, pp. 29 – 78 of [2].
- [21] M. Rajthen and A. Weiermann: Proof-theoretic investigations on Kruskal’s theorem, *Annals of Pure and Applied Logic* 60 (1993) 49 – 88.
- [22] K. Schütte: Kennzeichnung von Ordnungszahlen durch rekursive erklärte Functionen, *Meth. Ann.* 127 (1954) 15 – 32.
- [23] K. Schütte: *Proof Theory*, Springer-Verlag, 1977 (translated by J. N. Crossley).
- [24] J. R. Shoenfield: *Mathematical Logic*, Addison-Wesley, 1967.
- [25] Harold Simmons: *Derivation and Computation*, CUP 2000.
- [26] O. Veblen: Continuous increasing functions of finite and transfinite ordinals, *Trans. Amer. Math. Soc.* 9 (1908) 280 – 292.