HOW TO WELL ORDER FINITE TREES

H. JERVELL

REPORT No. 32, 2000/2001
ISSN 1103-467X
ISRN IML-R- -32-00/01- -SE

INSTITUT MITTAG-LEFFLER
THE ROYAL SWEDISH ACADEMY OF SCIENCES

How to well order finite trees

Herman Ruge Jervell
University of Oslo

June 7, 2001

Contents

1 The two traditions

There have been many investigations on how to build up ordinals from below — either since the ordinals are fundamental mathematical objects or for use in proof theoretic investigations. We shall here talk of two traditions. The first and larger one goes back to Veblen [12] and has as high points Bachmann [2], Schütte [6] and Buchholz [3]. The other tradition is smaller. The main papers are by Ackermann [1], Schütte [5] and a sequence of papers by Takeuti [7] [8] [9] [10] [11].

In the first tradition we give names for large ordinals. The names are built up by stages. Say we have come to stage α. Then we can at stage $\alpha + 1$ throw in names for some of the first ordinals not named so far. This was done systematically by Veblen [12] using terms built up from normal (i e continuous and increasing) functions of ordinals. Then the enumerating functions of fix points is also normal and give names for ordinals not named so far. This has later been pushed much further with the work of Bachmann [2]. Two drawbacks here are

- the same ordinal may have many names
- the reference of a naming term depends heavily on the basic functions we use

In this paper we'll give an exposition of the second tradition. There we use finite trees as representing ordinals. A remarkable thing is that we have unique names — two different trees represent different ordinals. Bachmann, Schütte and Takeuti consider slightly different finite trees

Ackermann Ordinary finite trees

Schütte Possibly infinite trees but with finite support.

Takeuti Ordinary finite trees where the nodes are labeled with well ordered labels.

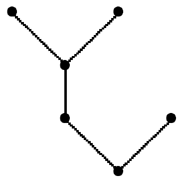
We shall consider each of this in turn. But note that we have done some simplifications

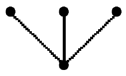
- All three of them had a term system with both an additive operation + and a tree constructor. We have taken away the +.
- There are two ways of building a tree from its immediate subtrees — as sequences or as multisets (bags). For the present exposition we have used sequences also for Takeutis ordinal diagrams.

The additive operation is crucial for the first tradition, but does not play a similar role here. We get to the same ordinals and the addition of + makes the proofs less perspicuous. The use of + is first an advantage when we want to simulate operation on proofs with the finite trees. The use of multisets instead of sequences makes trees equal if they are equal up to permutation of the branches. This makes the equality of trees slightly more complicated. On the other hand it may be an advantage when we want to compare the ordinals assigned to trees with some other ordinal notation. With multisets the proofs below go through with minimal changes.

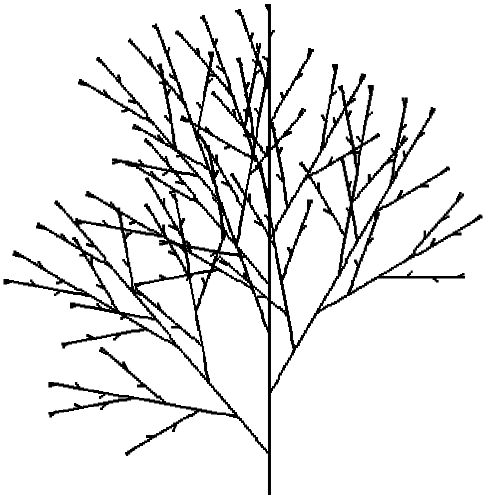
2 Finite trees

We start with finite trees. They can be like below





They may even be like



The smallest tree is the empty tree ε which we write as

$$\bullet$$

To each finite tree A there is the sequence of immediate subtrees of A — denoted by $\langle A\rangle$. The empty tree ε has as its sequence of immediate subtrees $\langle\varepsilon\rangle$ the empty sequence. We now come to Ackermanns ordering $<$ of finite trees. The equality is the ordinary equality of trees. As usual we define $>$, $\leq$ and $\geq$.

Between trees and sequence of immediate subtrees we shall define four orderings

$\mathbf{A < B}$**:** The ordering to be defined

$\mathbf{A \leq \langle B\rangle}$**:** $\exists B' \in \langle B\rangle \,.\, A \leq B'$

$\mathbf{\langle A\rangle < B}$**:** $\forall A' \in \langle A\rangle \,.\, A' < B$

$\mathbf{\langle A\rangle < \langle B\rangle}$**:** The lexicographical ordering of the two sequences

The lexicographical ordering is the ordering of finite sequences where we compare in order of importance

- the length of the two sequences
- the rightmost elements which are different

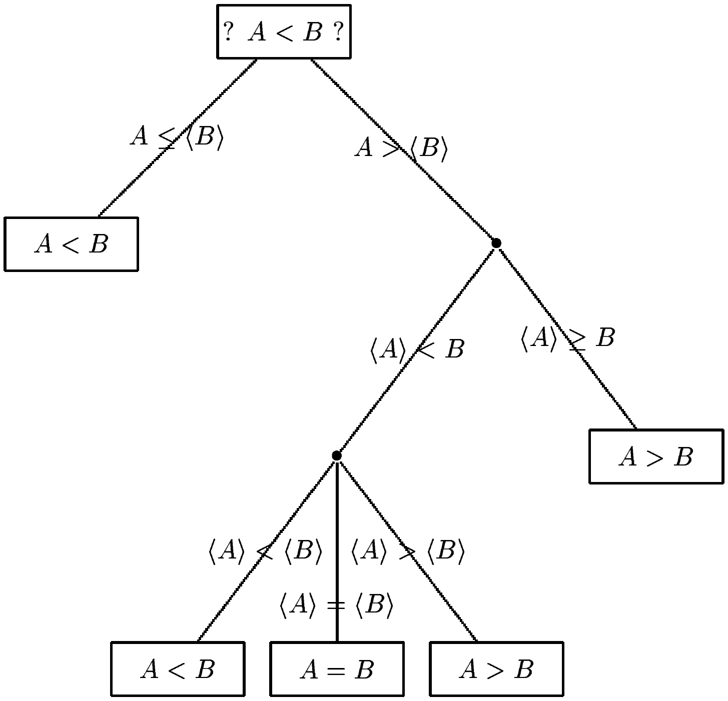
We then have the following simple inductive definition of the ordering of trees

$$\boxed{A < B \Leftrightarrow A \leq \langle B\rangle \,.\vee.\, \langle A\rangle < \langle B\rangle \wedge \langle A\rangle < B}$$

We observe that the empty tree is the smallest tree since the sequence of its immediate subtrees is empty — and for any other tree B we obviously have

$$\langle \varepsilon \rangle < \langle B \rangle \wedge \langle \varepsilon \rangle < B$$

The first more complicated observation is that the ordering is a total and decidable ordering. This is seen by writing down the decision tree for $A < B$. Note that in the choices of the decision tree we use that the ordering is total for smaller trees.



The proof that the ordering is total and decidable is by induction over the height of the tree. We also use height induction to prove that the ordering is transitive. So assume that

$$A < B \wedge B < C$$

and that we have transitivity for all triples of trees of smaller height.

We have cases according to how $B < C$ is proved and subcases according to how $A < B$ is proved. We have

- $B \leq \langle C \rangle$: by induction $A < \langle C \rangle$which gives $A < C$
- $\langle B \rangle < \langle C \rangle \wedge \langle B \rangle < C$
  - $A \leq \langle B \rangle$: use $A \leq \langle B \rangle \leq C$ to get $A < C$

- $\langle A\rangle < \langle B\rangle \wedge \langle A\rangle < B$. We then proceed
  - Use $\langle A\rangle < \langle B\rangle < \langle C\rangle$ to get $\langle A\rangle < \langle C\rangle$
  - Use $\langle A\rangle < B < C$ to get $\langle A\rangle < C$

  which also gives $A < C$

From the definition we immediately get that the immediate subtrees are smaller

$$B \in \langle A\rangle \Rightarrow B < A$$

There is a connection between embeddings of trees and our ordering. Let us write

$$A \prec B$$

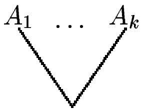
for A can be embedded into B. An elementary induction over the height of the trees show that

$$A \prec B \Rightarrow A < B$$

Another simple property is monotonicity

$$A < B \Rightarrow \mathcal{T}[A] < \mathcal{T}[B]$$

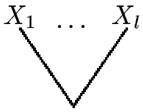
Here $\mathcal{T}[A]$ denotes the tree $\mathcal{T}$ where we have inserted A at some point, and $\mathcal{T}[B]$ the tree where we have inserted B at the same point. To get this and more we introduce the generating functions of a tree. Say we have a tree A with immediate subtrees $A_1, \ldots, A_k$



There are then two types of generating functions of A. The first is defined for any $l < k$ and is the function which to trees

$$X_1, \ldots, X_l$$

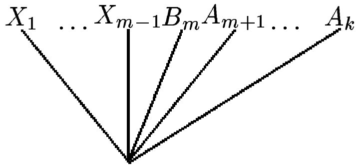
give the tree



So we are allowed to build trees with smaller branching than k. The next type of generating function is slightly more complicated. Then we are given a number $m \leq k$ and a tree $B_m < A_m$. We then have the function which to trees

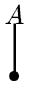
$$X_1, \ldots, X_{m-1}$$

give the tree



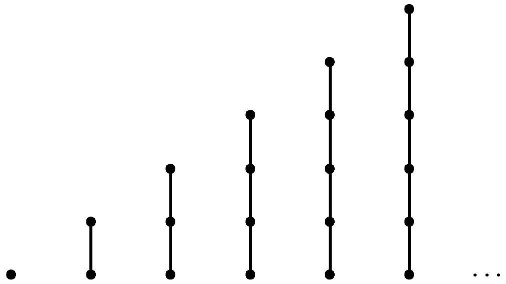
Theorem 1 *Given a finite tree A. The set of trees we get from the immediate subtrees $\langle A \rangle$ and close them under the generating functions give a set of trees cofinal in A i.e. for any tree $B < A$ we can find a tree in the set which is $\leq B$.*

This proves again that the empty tree is the smallest tree. Furthermore we can show that the successor of the tree A is



This tree have only as generating functions constants $\leq A$ — and therefore any smaller tree must also be $\leq A$.

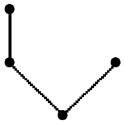
Then we are ready to identify some ordinals. The finite numbers are given by



The first infinite ordinal is ω and this is the tree

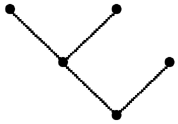
It has only one generating function — the function which to X gives

and we get the cofinal set starting with the empty tree. We know how to make the successor. To make a further jump we look at the tree

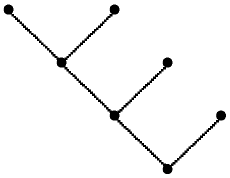


Now we have as generating functions the successor and the constant ω

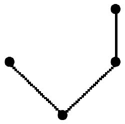
And we get the ordinal $\omega \cdot 2$. Further calculation gives the ordinal ω^2 as



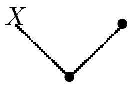
And ω^3



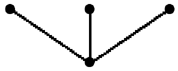
We are ready for ω^ω. This is the tree



It has as generating functions both the successor function and the function which to tree X gives



There are a number of important ordinals coming up. The first major jump is ϵ_0. It is simply the tree



And we have Γ_0

3 The well ordering of the finite trees

We are now ready to prove that our linear ordering of the finite trees is a well ordering. Our proof here is a translation of Takeutis proof of the well ordering of ordinal diagrams into our setting. This will also pave the way for the more complicated proofs of well ordering of ordinal diagrams that will follow.

There are two crucial concepts involved in the proof. First we say that A is accessible, $A \in \mathcal{A}$, if there are no infinitely descending sequence of trees starting from A. Secondly we say that A is $\mathcal{T}$-accessible, $A \in \mathcal{T}$, if all the immediate subtrees of A are accessible ie $\langle A \rangle \subseteq \mathcal{A}$.

Lemma 1 *If $A \notin \mathcal{A}$, then either $A \in \mathcal{T} - \mathcal{A}$ or there is a subtree B of A with $B \in \mathcal{T} - \mathcal{A}$.*

For assume $A \notin \mathcal{T}$. Then there is an immediate subtree A_0 of A with $A_0 \notin \mathcal{A}$. If this A_0 is not in $\mathcal{T}$, then there is an immediate subtree A_1 of A_0 with $A_1 \notin \mathcal{A}$. So we go on and get a sequence of subtrees $A > A_0 > A_1 > \cdots$ where none are accessible. The ultimate subtree is the empty tree ε which is accessible. So the sequence must stop with either A or a subtree B which is in $\mathcal{T} - \mathcal{A}$.

Lemma 2 *If $A \notin \mathcal{A}$, then there is an infinite descending sequence of trees from $\mathcal{T}$*

We use the lemma above to first pick out an infinite descending sequence where the first element is in $\mathcal{T} - \mathcal{A}$. Then we look at the second element. If this is not in $\mathcal{T} - \mathcal{A}$, then a subtree of it will be. Make this the second element in the wanted infinite descending sequence. And so we go on for the third element, fourth element etc. Here we use dependent choice and the choice depends on trees being accessible or not. This is the place in the proof where we really use proof theoretic strength. Most of the rest can be done by simple induction formulated in say Peano arithmetic.

Lemma 3 *If $A > B$ and $A, B \in \mathcal{T} - \mathcal{A}$, then $\langle A \rangle > \langle B \rangle$.*

For assume $A > B$ and $A, B \in \mathcal{T} - \mathcal{A}$. If $\langle A \rangle \geq B$, then since $A \in \mathcal{T}$ we must have $B \in \mathcal{A}$. Therefore $\langle A \rangle < B$. Since $A > B$ we get $\langle A \rangle > \langle B \rangle \wedge A > \langle B \rangle$.

Theorem 2 *All trees are accessible.*

If we had an infinite descending sequence

$$A_0 > A_1 > A_2 > \cdots$$

we could assume that all the elements are $\mathcal{T}$-accessible and that the following sequence is also descending

$$\langle A_0 \rangle > \langle A_1 \rangle > \langle A_2 \rangle > \cdots$$

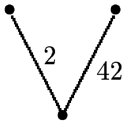
The length of each sequence cannot increase. Therefore we can — after having cut off some initial elements — assume that all the lengths are equal. Furthermore — again after having cut off some initial elements — there must be a number k such that the k'th elements of each sequence is decreasing

$$A_{0,k} > A_{1,k} > A_{2,k} > \cdots$$

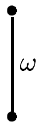
Here $A_{i,k} \in \langle A_k \rangle$. But then $A_{0,k} \notin \mathcal{A}$ and $A_0 \notin \mathcal{T}$. Contradiction.

4 Trees with finite support

In [5] Kurt Schütte generalized the Ackermann ordering to trees with finite support. These are well founded trees where at each node there is only a finite number of immediate subtrees but where there may be gaps in the branching. Here is a tree where we have two immediate subtrees — one above branch 2 and the other at branch 42.



The first interesting tree is the one where we have an immediate subtree at branch ω



Schütte invented the "Klammersymbole" as a notation to describe such trees. We remember that the empty tree ε corresponds to the ordinal 0. Then the first tree is written as

$$\begin{vmatrix} 2 & 42 \\ 0 & 0 \end{vmatrix}$$

and the second as

$$\begin{vmatrix} \omega \\ 0 \end{vmatrix}$$

We can go further and replace the symbols 2, 42, ω with the appropriate Klammersymbole etc. We leave this development to the reader.

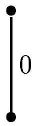
There is a simple inductive definition of the Klammersymbole. To get it we start with the empty tree ε and use finite partial functions. We write

$$X \rightharpoonup Y$$

for the finite partial functions from X to Y. The Klammersymbole is given as

$$\boxed{\mathcal{S} = \{\varepsilon\} \cup (\mathcal{S} \rightharpoonup \mathcal{S})}$$

Since we use partial functions this is an OK inductive definition. We start with ε. Then we get an extra element by the function which takes ε to ε. This is the tree



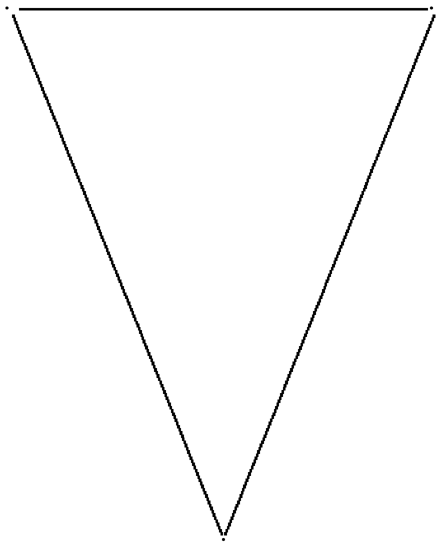
which we recognize as the ordinal 1. From there we can proceed and get many more trees.

5 Labeled finite trees

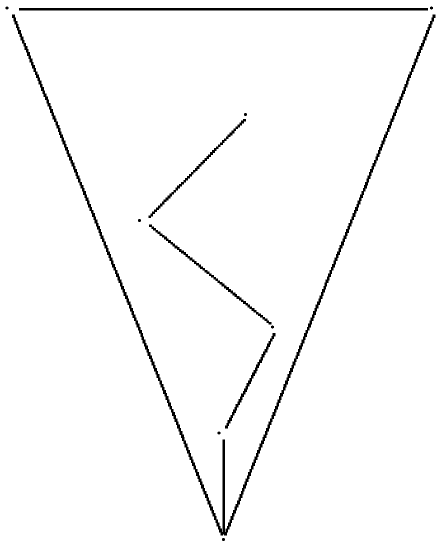
We can consider Takeutis ordinal diagrams as finite trees where the nodes are labeled. The labels are taken from a well ordered set given in advance. There is no attempt of making the labels as ordinal diagrams etc. Instead we shall see that we get interesting new trees where the labels are from simple infinite well ordered sets like ω or $\omega + 1$. So we start with a well ordered set I and make as ordinal diagrams $\mathcal{O}(I)$ the finite trees labeled with I.

Takeuti developed first the ordinal diagrams of finite order [7] [8]. These are $\mathcal{O}(I)$ where the labeling set I is finite. Later he showed how the theory can be generalized to ordinal diagrams with infinite order [9]. For the first important application there — proving consistency of Π^1_1-CA — he used a labeling set just above ω.

To get to Takeutis theory we first have to explain the concept "path satisfying i-gap condition". Remember that we consider finite trees where the nodes are labeled with elements from a well ordered set I. We first write down the tree



Within the tree we have a path



This path satisfies the i-gap condition if

- the topmost node in the path has label i
- all other nodes in the path has labels $> i$

We are now ready to define

i-section: To a tree A the i-section $\langle A \rangle_i$ is the sequence of subtrees of A above paths satisfying i-gap condition

Next label: In the context of some trees we let $i+$ be the smallest label $> i$ if such a label exists in the trees, or ∞ otherwise.

Describing a node: A node is given by its label i and the sequence of its immediate subtrees $\langle A \rangle$. We use the notation $(i, \langle A \rangle)$.

And get the Takeuti-orderings $<_i$ and $<_\infty$. They are given by

1. $A <_i B \Leftrightarrow A \leq_i \langle B \rangle_i \,.\vee.\, A <_{i+} B \wedge \langle A \rangle_i <_i B$
2. $(i, \langle A \rangle) <_\infty (j, \langle B \rangle) \Leftrightarrow i < j \,.\vee.\, i = j \wedge \langle A \rangle <_i \langle B \rangle$
3. The main Takeuti-ordering is $<_0$

To familiarize us with the definition we first look at trees where the only label is 0. The smallest tree is of course

$$0$$

We see that the definition then simplifies to the usual Ackermann ordering. The next observation is that the tree

$$n$$

is larger than all the trees built up from labels $< n$.

As for the Ackermann ordering we prove that the Takeuti orderings are total, decidable, transitive. We leave to the reader to make the small changes needed for the Takeuti case.

It is hard to get a feeling for the Takeuti-ordering. The ordinal diagrams obviously goes beyond the Ackermann-ordering. The finite trees corresponds to trees with the same label at all nodes. If we have different labels in the trees, it may be quite complicated. One may try it by hand — or use the Haskell code in the last chapter in this report.

In an attempt to understand the ordering Takeuti has introduced a theory of approximation [10] [11]. Let me explain the simplest part of it.

1. A part of an ordinal diagram is i-active if it can be reached from the root only going through labels $\geq i$. We do not count the labels in the part itself — so the ordinal diagram is i-active for all i.
2. The i-value of an ordinal diagram A, written $\nu(i, A)$, is the maximum label which is i-active.
3. A non-trivial ordinal diagram has always an i-value. For $i = 0$ $\nu(0, A)$ is the maximum label in the tree. For $i > 0$ $\nu(i, A)$ may be less than the maximum label.

4. Among the i-active sub diagrams of A we look at the sub diagrams $\mathcal{S}$ which are i-active and with label $\nu(i, A)$ at the bottom. The i-approximation of A, $\mathrm{App}(i, A)$, is the ordinal diagram $(i, \langle B\rangle)$ where $\langle B\rangle$ is the sequence of ordinal diagrams we get from $\mathcal{S}$. Note that $\mathcal{S}$ are the sub diagrams with label $\nu(i, A))$ at bottom, not label i at bottom.

Then by a reasonably straightforward induction we have

Theorem 3 *For ordinal diagrams A and B*

- *If $\nu(i, A) < \nu(i, B)$, then $A <_i B$.*
- *If $\nu(i, A) = \nu(i, B) = j$ and $App(i, A) <_j App(i, B)$, then $A <_i B$.*

This shows that in the ordering $<_0$ of ordinal diagrams we have in order

- All ordinal diagrams with only label 0 — ie the Ackermann trees
- All ordinal diagrams with labels ≤ 1 and containing at least one label 1.
- ...
- All ordinal diagrams with labels $\leq n$ and containing at least one label n.
- ...

6 The well ordering of ordinal diagrams

The proof is a generalization of the proof in the Ackermann case. We need two crucial notions — one generalizing accessibility and the other generalizing $\mathcal{T}$-accessibility.

For the first one let X be a set of trees. Then $B \in \mathcal{A}^X_<$ iff there are no $>$-descending sequence of trees starting with B and contained within X.

For the second we define

- $\mathcal{T}_0 =$ all trees
- $A \in \mathcal{T}_{i+1} \Leftrightarrow A \in \mathcal{T}_i \wedge \langle A\rangle_i \subseteq \mathcal{A}^{\mathcal{T}_i}_{<_i}$
- $\mathcal{T}_\lambda = \bigcap_{\kappa<\lambda} \mathcal{T}_\kappa$
- $\mathcal{T}_\infty = \mathcal{T}_\alpha$ where α is larger than all labels I

The following are simple properties for the $\mathcal{T}_i$

- if $i < j$, then $\mathcal{T}_i \supseteq \mathcal{T}_j$
- if $A \in \mathcal{T}_{i+1}$ and $a \in \langle A\rangle_i$, then $a \in \mathcal{T}_i$

For the last assume $A \in \mathcal{T}_{i+1}$ and $a \in \langle A \rangle_i$ and $a \notin \mathcal{T}_i$. But then there is $j < i$ and $b \in \langle a \rangle_j$ starting an infinite $<_j$-descending sequence in $\mathcal{T}_j$. Now note that we have also $b \in \langle A \rangle_j$ and hence $A \notin \mathcal{T}_{j+1}$. Contradiction.

Assume that we have a tree A^0_0 which starts an infinite $>_0$-descending sequence

$$A^0_0 >_0 A^0_1 >_0 A^0_2 >_0 \cdots$$

We shall construct the Takeuti-matrix consisting of elements

$$A^i_n$$

where i is a label or ∞ and n a natural number. The elements will have the properties

- for each i, the rows A^i_n are infinite $>_i$-descending
- the elements in row i are all in $\mathcal{T}_i$
- in each column n, there are only a finite number of distinct trees

The construction will go row for row starting with the top row, and to get from one row to the next our construction is quite similar to the Ackermann case. At the end — after having constructed row ∞ — we get a contradiction in a way which is also similar to the Ackermann case.

So assume we have the top row constructed. We have

$$A^0_0 >_0 A^0_1 >_0 A^0_2 >_0 \cdots$$

and they are trivially all in $\mathcal{T}_0$ which consists of all trees. We now follows the Ackermann case in constructing an infinite sequence

$$A^1_0 >_0 A^1_1 >_0 A^1_2 >_0 \cdots$$

where all the trees are in $\mathcal{T}_1$. We do this starting from left in the row

$$A^0_0 >_0 A^0_1 >_0 A^0_2 >_0 \cdots$$

and finding the first tree not in $\mathcal{T}_1$. We can find a subtree of it which is in $\mathcal{T}_1$. Then continue as in the Ackermann case until the whole row is in $\mathcal{T}_1$. This becomes

$$A^1_0 >_0 A^1_1 >_0 A^1_2 >_0 \cdots$$

For later use we note that *the first place where the two rows differs we have a subtree in row 1 of the tree in row 0.* Now we go back to the Ackermann case. Similar to the proof there we want to prove that row 1 is also $>_1$-descending. So we have for each n

$$A^1_n >_0 A^1_{n+1}$$

and they are both elements in $\mathcal{T}_1 - \mathcal{A}^{\mathcal{T}_0}_{<_0}$. If $A^1_{n+1} \leq_0 \langle A^1_n \rangle_0$, then $A^1_n \notin \mathcal{T}_1$. So $A^1_{n+1} >_0 \langle A^1_n \rangle_0$ and hence $A^1_n >_1 A^1_{n+1}$. But then

$$A^1_0 >_1 A^1_1 >_1 A^1_2 >_1 \cdots$$

and all elements are in $\mathcal{T}_1 - \mathcal{A}^{\mathcal{T}_0}_{<_0}$. We can then proceed to the next rows and get for each n

- $A^n_0 >_n A^n_1 >_n A^n_2 >_n \cdots$
- $A^n_k \in \mathcal{T}_1$
- the first element where row n and $n+1$ differ the element of row $n+1$ is a subtree of the element of row n. We call this element in row n for the critical element of row n.

Takeuti saw in [9] that the construction could proceed to rows labeled with limit ordinals. Then the crucial observation is the one about the critical elements in a row having a subtree of it immediately below. We show that in all columns there are only finitely many changes in the elements. Assume not. Then there is a leftmost column — say column k — where there are infinitely many changes. From some row on — say row n — then there are no changes in the columns to the left of k. This means that in column k from row n downwards the elements will either be the same as the element above or a subtree of it. The case of subtrees will occur infinitely many times. This gives a contradiction and we can conclude that in each column there are only finitely many changes. But then it is easy to go the limit row. In each column the elements will be constant from some row on — and we take that element to be its limit.

We let row ∞ be the row for the smallest ordinal larger than all labels. We have

$$A^\infty_0 >_\infty A^\infty_1 >_\infty A^\infty_2 >_\infty \cdots$$

Furthermore the elements will be in $\mathcal{T}_i$ for any i.

Now look at the definition of $<_\infty$. From some k onwards in row ∞ all labels must be the same - say label i. We can write it as

$$(i, \langle A_k \rangle) >_\infty (i, \langle A_{k+1} \rangle) >_\infty (i, \langle A_{k+2} \rangle) >_\infty \cdots$$

Observe now that the elements are all in each $\mathcal{T}_j$. In particular they are in $\mathcal{T}_{i+1}$ and the elements in the i-sections are all in $\mathcal{T}_i$ by the simple property of the $\mathcal{T}_i$ we started our proof with.

We get a descending sequence of the i-sections

$$\langle A_k \rangle >_i \langle A_{k+1} \rangle >_i \langle A_{k+2} \rangle >_i \cdots$$

and as before we can pick out a number l so that elements l in the i-sections are descending

$$A_{k,l} >_i A_{k+1,l} >_i A_{k+2,l} >_i \cdots$$

Remember that all the elements in the sequence are in $\mathcal{T}_i$. But then $(i, \langle A_k \rangle) \notin \mathcal{T}_{i+1}$. Contradiction.

We conclude that

- the Takeuti-matrix does not exist
- all the Takeuti-orderings are well orderings

7 Ordinal diagrams are dilators

In the ordinal diagrams we have as parameters the set of labels. It is worthwhile to note how the parameters are used in the construction. To compare two ordinal diagrams A and B we only need to take into account the ordering relation used between the labels of A and B. Hence an ordinal diagram is a functor from the category of linear orders to the category of linear orders. The functor preserves

Direct limits: Since the ordinal diagrams are finite trees.

Pull backs: Since we have uniqueness of the trees. Different trees are different in the ordering.

Well ordering: This is Takeutis result that the ordinal diagrams are well ordered.

This gives

Theorem 4 *Ordinal diagrams are dilators. [4]*

8 Haskell code

We have written down codes in Haskell for the Ackermann- and the Takeuti-ordering. It may be useful to see it done, but note that the code is not optimized.

```
data Ack    =   A [Ack]

eq      ::  Ack  ->  Ack  ->  Bool

eq (A[]) (A[])      =  True
eq (A[]) (A(y:ys))  =  False
eq (A(x:xs)) (A[])  =  False
eq (A(x:xs)) (A(y:ys))  =  (eq  x  y)&& (eq  (A xs)(A ys))
```

```
lexi        :: [Ack] -> [Ack] -> Bool

lexi x y    |   (length x) < (length y)     = True
            |   (length x) > (length y)     = False

lexi (x:xs) (y:ys)      |   (less x y)  = True
                        |   (less y x)  = False
                        |   (eq x y)    = lexi (xs) (ys)


lexi [][]       =  False

lesseq x y      = (less x y) || (eq x y)

greater x y     = less y x


less (A x) (A y)    = ( any (lesseq (A x)) y ) ||
                         ((lexi x y) && (all (greater (A y)) x))

nul         =  A []

one          =  A [nul]

omega           =  A [nul,nul]

suk x           =  A[x]

num 0           =  nul

num n           =  suk (num (n-1))

epsilonnull     =  A[nul,nul,nul]

gammanull       =  A[nul,nul,nul,nul]


data    Tak =   T (Int,[Tak])

eq      ::  Tak  ->  Tak  ->  Bool  ---  equality of trees

eq (T(x,[])) (T(y,[]))      =  ( x == y )

eq (T(_,[])) (T(_,(y:ys)))  =  False
```

```
eq (T(_,(x:xs))) (T(_,[])) = False

eq (T(u,(x:xs))) (T(v,(y:ys)))  | ( u /= v) =  False
                                | eq x y    =  eq (T(u,xs)) (T(v,ys))


lexi    :: Int -> [Tak] -> [Tak] -> Bool  --- lexicographical
                                          ---  ordering left to right

lexi i x y  |   (length x) < (length y)     = True
            |   (length x) > (length y)     = False

lexi i (x:xs) (y:ys)    |   (less i x y)    = True
                        |   (less i y x)    = False
                        |   (eq x y)        = lexi i (xs) (ys)


lexi  i [][]        =  False

lesseq i x y        = (less i x y) || (eq x y)

greater i x y       = less i y x


section     ::  Int -> Tak -> [Tak]

section i (T(j,x))  | (i == j)  = x

section i (T(j,(x:xs))) | (i < j)   = []
                        | (i > j)   = (section i x) ++ (section i (T(j,xs)))

strip       ::  Tak -> [Int]  ---  finds the list of indices used

strip (T(i,[]))     = [i]

strip (T(i,(x:xs))) = [i] ++ (strip x) ++ (strip (T(i,xs)))

maks        ::  Tak -> Tak -> Int    ---  finds maximal index in
                                     ---  two ord.diagrams

maks x y    = max (maximum (strip x)) (maximum (strip y))


less k (T(i,x)) (T(j,y))    =   ( any (lesseq k (T(i,x))) y ) ||
                                 ((lessnext k (T(i,x)) (T(j,y)))
```

```
                                   && (all (greater k (T(j,y))) x))


lessnext k (T(i,x)) (T(j,y))| (k <= (maks (T(i,x)) (T(j,y))))   =
                                        less (k+1) (T(i,x)) (T(j,y))
                            | (i < j)        = True
                            | (i > j)        = False
                            | otherwise      = lexi i x y


zero    =   T (0,[])

suk x   =   T (0,[x])

omega   =   T (0,[zero,zero])
```

References

[1] Ackermann, W., *Konstruktiver Aufbau eines Abschnitts der zweiten Cantorschen Zahlenklasse*, Math. Z., 53, 1951.

[2] Bachmann, H., *Die Normalfunktionen und das Problem der ausgezeichneten Folgen von Ordnungszahlen*, Vierteljahrschr. Natutf. Ges. Zürich 95, 1950.

[3] Buchholz, W., *Eine Erweiterung der Schnitteliminationsmethode*, Habilitationsschrift, München, 1977.

[4] Girard, J-Y, *A survey of Π^1_2-logic. Part I: Dilators*, Ann.Math.Logic 21, 1981, pp 75-219.

[5] Schütte, K., *Kennzeichnungen von Ordnungszahlen durch rekursiv erklärte Funktionen*, Math. Ann. 127, 1955, pp 15-32.

[6] Schütte, K., *Eine konstruktives System von Ordinalzahlen I*, Arch. math. Logik Grundlagenforsch. 11, 1968.

[7] Takeuti, G., *Ordinal diagrams*, Journ. Math. Soc. Japan, 9, 1957.

[8] Takeuti, G., *Ordinal diagrams II*, Journ. Math. Soc. Japan, 12, 1960.

[9] Takeuti, G., *Consistency proofs of subsystems of classical analysis*, Ann. Math. 86, 1967, pp 299-348.

[10] Takeuti, G., *Proof theory*, North-Holland, 1975.

[11] Takeuti, G., and Mariko Yasugi, *Fundamental sequences of ordinal diagrams*, Comment.Math. Univ.St.Pauli XXV-I, 1976, pp 1-80.

[12] Veblen, O., *Continuous increasing functions of finite and transfinite ordinals*, Trans. Amer. Math. Soc. 9, 1908.