Ordering finite labeled trees

Herman Ruge Jervell

December 6, 2007

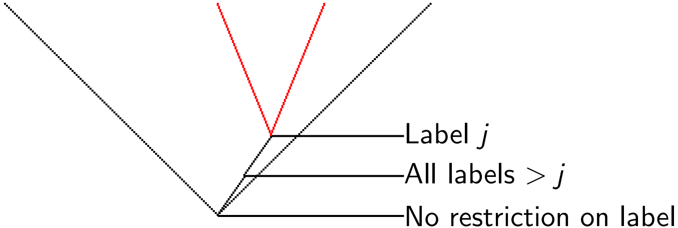
Goal

- Given a well ordered set of labels $\Lambda = 0, 1, 2, \ldots$
- We define a new ordering of finite trees with labels at the nodes
- We prove that the ordering is a well order

The ordering of finite trees is the initial segment consisting of finite trees labeled with only 0.

Subtrees

j-subtrees where j is a label. Write $S \subset_j T$



- $S \subset_i T \subset_j U \wedge i < j \Rightarrow S \subset_i U$

$\langle S \rangle_i$ is the finite sequence of i-subtrees of S

Orderings $<_i$ and $<_\infty$

$$S <_j T \Leftrightarrow S \leq_j \langle T \rangle_j \vee (\langle S \rangle_j <_j T \wedge S <_{j+} T)$$
$$S <_\infty T \Leftrightarrow \text{lexicographical ordering}$$

- $\leq_j$ means either ordinary $=$ or $<_j$
- $S \leq_j \langle T \rangle_j$: There exists a j-subtree T_0 of T with $S \leq_j T$
- $\langle S \rangle_j < T$: For all j-subtrees S_0 of S , $S_0 <_j T$
- $j+$ is the smallest label in S and T larger than j if it exists, else it is ∞
- The lexicographical ordering is such that we compare in priority
  - The labels at the root of S and the root of T
  - For the same label i: The lengths of $\langle S \rangle_i$ and $\langle T \rangle_i$
  - The rightmost place where the two sequences differ in the $>_i$-ordering

Transitivity

Assume $\mathbf{A} <_j \mathbf{B} <_j \mathbf{C}$. Then we have cases

- $\mathbf{B} \leq_j \langle \mathbf{C} \rangle_j$: Then $\mathbf{A} <_j \mathbf{B} \leq_j \langle \mathbf{C} \rangle_j$, and by induction $\mathbf{A} \leq_j \langle \mathbf{C} \rangle_j$ and $\mathbf{A} <_j \mathbf{C}$
- $\mathbf{B} <_{j+} \mathbf{C}$ and $\langle \mathbf{B} \rangle_j <_j \mathbf{C}$: Then
  - $\mathbf{A} \leq_j \langle \mathbf{B} \rangle_j$: Then $\mathbf{A} \leq_j \langle \mathbf{B} \rangle_j <_j \mathbf{C}$ and by induction $\mathbf{A} <_j \mathbf{C}$
  - $\mathbf{A} <_{j+} \mathbf{B}$ and $\langle \mathbf{A} \rangle_j <_j \mathbf{B}$: Then we have $\mathbf{A} <_j \mathbf{C}$ from
    - $\mathbf{A} <_{j+} \mathbf{B} <_{j+} \mathbf{C}$ which gives by induction $\mathbf{A} <_{j+} \mathbf{C}$
    - $\langle \mathbf{A} \rangle_j <_j \mathbf{B} <_j \mathbf{C}$ which gives by induction $\langle \mathbf{A} \rangle_j <_j \mathbf{C}$

For $j = \infty$

- The labels at (least two of) the roots are different. Then $\mathbf{A} <_\infty \mathbf{C}$
- The labels are the same. Then we compare i-subtrees and by induction $\mathbf{A} <_\infty \mathbf{C}$

We may have $A <_i A$.

Bad and minimal trees

- S is i-bad — There exists infinitely descending $<_i$-sequence starting with S.
- S is i-minimal — For all $j \leq i$: No element of $\langle S \rangle_j$ is j-bad.

We prove that there are no 0-bad trees and in the proof we use the i-minimality in an essential way.

Step 1 – Lexicographical ordering

As a warming up - well ordering is closed under extension by lexicographical order.

- Assume $S, T, \ldots$ are wellordered — so none of them are bad
- Assume $\langle U, V, \ldots, W \rangle$ is bad
- Consider the infinitely descending sequence of sequences starting with $\langle U, V, \ldots, W \rangle$
- From some stage of — the sequences must have the same length
- The rightmost element which changes infinitely often gives a bad sequence
- Contradiction

Step 2 — Only one label 0

This is really the case of wellordering of finite trees. But we prove it here by a minimal badness argument.

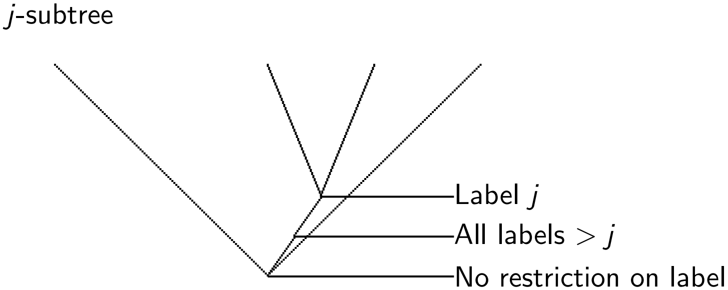
- Assume S is bad
- If S is not minimal, then one of its immediate subtrees are bad
- The topmost subtrees — being the empty trees — are minimal but not bad
- By going to subtrees we get a minimal bad tree T_0
- We then look at the second element, T, in the bad sequence starting with T_0
- The immediate subtrees of T are all $<_0$ T_0
- By going to subtrees above T we get a minimal bad tree T_1 with $T_0 >_0 T_1$
- Continuing in this way we get a sequence $T_0 >_0 T_1 >_0 T_2 >_0 \cdots$ of minimal bad trees

Step 2 — Label 0 — Conclusion

Consider the ordering:
$S <_0 T \Leftrightarrow S \leq_0 \langle T \rangle_0 \vee (\langle S \rangle_0 <_0 T \wedge S <_\infty T)$

- The first part was used when we went to subtrees to find minimal bad trees
- When we have minimal bad trees we must use the second part of the definition
- This means that we have also $T_0 >_\infty T_1 >_\infty T_2 >_\infty \cdots$
- This is the case of lexicographical ordering that we treated in step 1
- Contradiction

The gap condition



$$S \subset_j T \wedge i < j \Rightarrow \langle S \rangle_i \subseteq \langle T \rangle_i$$

Minimal

A tree S is i-minimal if for all $j \leq i$ no element of $\langle S \rangle_j$ is $<_j$-bad

Assume we have a sequence of i-bad i-minimal trees. Then our subtree construction gives a sequence of $i+1$-bad $i+1$-minimal trees. We use that the $i+1$-subtree construction does not produce any new j-subtree where $j \leq i$.

For the complexity — note that we must decide whether we have a sequence of $i+1$-bad $i+1$-minimal trees.

Step 3 — Finite number of labels

- Assume S is 0-bad
- As in step 2 we construct a 0-minimal, 0-bad sequence $S_0^0 >_0 S_1^0 >_0 S_2^0 >_0 \cdots$
- As in step 2 this sequence is also 1-bad
- We then construct a 1-minimal, 1-bad sequence $S_0^1 >_1 S_1^1 >_1 S_2^1 >_1 \cdots$
- This sequence is then also 2-bad, and we continue to construct minimal, bad sequences row for row
- Row ∞ consists of trees which are ∞-bad and for all labels i they are i-minimal
- We find a label k and an element S_l^∞ with a k-bad element in $\langle S_l^\infty \rangle_k$
- This contradicts that S_l^∞ is k-minimal

Minimal — complexity

Look at our constructions of minimal rows

- Row 0: We quantify over all trees. We use Π^1_1 to distinguish between the bad and the not bad trees.
- Row 1: We quantify over all 0-minimal trees.
- Row $n+1$: We quantify over all n-minimal trees.

Step 4 — Infinite number of labels

- We prove that in each column in the matrix S^i_j there are only finite number of changes
- Take the leftmost column j with an infinite number of changes
- From a row i of there are no changes in the columns to the left of column j
- From row i all changes in column j are by taking subtrees
- There cannot be an infinite number of changes in column j

But then it is straightforward to define rows for limit ordinals and our construction works as well for an infinite number of labels as long as the set of labels are well ordered.

The labeled trees are well ordered