Trees, ordinals and termination*

Nachum Dershowitz
Department of Computer Science
University of Illinois
Urbana, IL 61801
U.S.A.

The Weizmann Institute of Science
Rehovot 76100
Israel

nachum@cs.uiuc.edu

Know that one is the secret and source of all the cardinals.

— Abraham ibn Ezra (1153)

Abstract

Trees are a natural representation for countable ordinals. In particular, finite trees provide a convenient notation the predicative ones. Processes that transform trees or terms can often be proved terminating by viewing the tree or the tree representation of the term as an ordinal.

1 Introduction

Cantor invented the ordinal numbers

$$0, 1, 2, \ldots \omega, \omega+1, \ldots \omega 2, \ldots \omega n,$$
$$\ldots \omega^2, \ldots \omega^n, \ldots \omega^\omega, \ldots \omega \uparrow n, \ldots$$
$$\epsilon_0, \ldots \epsilon_0^{\epsilon_0}, \ldots \epsilon_1, \ldots \epsilon_{\epsilon_0}, \ldots, \text{etc.}$$

Each ordinal is larger than all preceding ones, and is typically defined as the set of them all:

$$\begin{aligned}
&\omega = \text{the set of natural numbers}\\
&\omega 2 = \omega \cup \{\omega + n | n \in \omega\}\\
&\omega n = \cup_{i<n}\ \omega i\\
&\omega^2 = \cup_{n\in\omega}\ \omega n\\
&\omega^n = \cup_{i<n}\ \omega^i\\
&\omega^\omega = \cup_{n\in\omega}\ \omega^n\\
&\omega \uparrow n = \cup_{i<n}\ \omega \uparrow i\\
&\epsilon_0 = \omega^{\epsilon_0} = \cup_{n\in\omega}\ \omega \uparrow n\\
&\epsilon_0^{\epsilon_0} = \omega^{\epsilon_0^2}\\
&\epsilon_1 = \cup_{n\in\omega}\ \epsilon_0 \uparrow n
\end{aligned}$$

The notation $\alpha \uparrow n$ represents a tower of n αs.

Infinite sets cannot provide a *notation* for ordinals. In Section 2 we look for bijections between natural classes of finite trees and initial segments of the countable ordinals.

Turing [1950] and Floyd [1967] suggested using ordinals for proving termination of programs. Earlier, in 1938, Gentzen used an ϵ_0 ordering to show that Peano Arithmetic is consistent, by showing the termination of a proof-tree transformation process. In Section 3, we give some examples of termination proofs that follow directly from tree representations of ordinals.

*Research supported in part by the U. S. National Science Foundation under Grants CCR-90-07195 and CCR-90-24271 and by a Meyerhoff fellowship at The Weizmann Institute of Science.

2 Trees as ordinals

Finite rooted trees correspond one-to-one with the ordinals up to ϵ_0: the one-node tree is 0; a tree with subtrees corresponding to $\alpha_1, \ldots, \alpha_n$ corresponds to the natural (commutative) sum $\omega^{\alpha_1} + \cdots + \omega^{\alpha_n}$. This ordering is natural in that trees are larger than their subtrees and replacing a subtree by a smaller one gives a smaller tree. Moreover, replacing a subtree with any finite number of smaller trees results in a smaller tree. Ordered this way, finite trees give all ordinals up to ϵ_0. (See [Dershowitz and Manna, 1979] for well-founded orderings of multisets.)

One can, alternatively, consider ordered trees and use the same interpretation, except for substituting noncommutative addition. This is not a bijection, but if, in addition, we insist that subtrees are listed in non-ascending order, then this is just another way of writing an ordinal less than ϵ_0 in Cantor normal form.

The standard correspondence between ordered trees and binary trees leads to the following interpretation of the latter: a tree is no smaller than either of its immediate subtrees; replacing one of the subtrees with one no larger, gives something no larger; if the left branch of tree t is larger than that of s and the whole of t is larger than the right branch of s, then t is larger than s. See Table 1 for some examples.

Going farther, one can associate the countable ordinals with infinite binary trees in which no path has infinitely many left turns. The empty binary tree is 0; if the right branch is infinite, the tree corresponds to the limit of the ordinals obtained by truncating the right branch at deeper and deeper points; otherwise, one gets $\omega^x + y$ for the binary tree whose left branch corresponds to x and right branch to y, unless $\omega^x + y = x$, in which case we add one. This is analogous to *tree ordinals* as defined in [Dennis-Jones and Wainer, 1983]: 0 is a tree ordinal; if α is then so is $\alpha + 1$; if α_n ($n \in \mathbf{N}$) are, then the ω-sequence $\alpha_0, \alpha_1, \ldots$ (representing their limit) also is.

Rational binary trees (trees with only finitely many different subtrees) can be represented finitely (as cyclic graph structures) and—with a natural extension of the ordering on finite trees—give all ordinals up to (and including) $\epsilon_{\epsilon_{\cdots}}$. The effect is essentially the same as allowing leaves to be labeled by trees (themselves having such leaves). A leaf containing something is larger than trees with smaller leaves and corresponds to the epsilon number indicated by the contents of the leaf. In general, a tree corresponds to $\omega^x + y$, where x is the ordinal corresponding to the left branch and y corresponds to the right branch, unless x is an epsilon number ϵ_z, in which case one gets $x + y - z$ ($-$ is ordinal difference). An infinitely bifurcating rational tree corresponds to the critical ordinal $\epsilon_{\epsilon_{\cdots}}$. See [Dershowitz and Reingold, 1992].

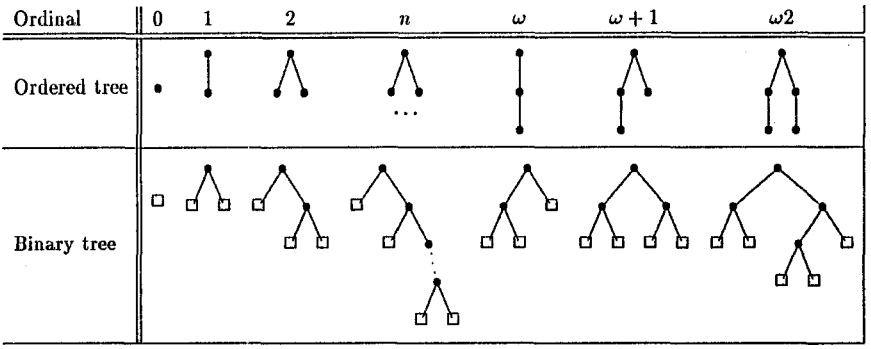
Labeled rooted trees also provide a natural notation for much larger ordinals. A *supertree* is a tree whose nodes can themselves be supertrees. Supertrees with identical roots are compared as were rooted trees above. But if tree s has a larger root than t, and s is larger than each of the subtrees of t, then s is larger than t. See Table 2.

Let $\phi^1(0), \phi^1(1), \ldots$ enumerate the epsilon numbers, and $\phi^\alpha(\beta)$ enumerate the fixpoints $\phi^\mu(\beta) = \beta$ that are common to all $\mu < \alpha$. (See [Schmidt, 1976].) Then we extend the mapping of trees to ordinals by making s greater than t if the root of s is greater than the root of t and s is greater than the subtrees of t. This maps a tree with root corresponding to α and subtrees $\beta_1, \ldots, \beta_n$ to $\phi^\alpha(\beta_1 + \cdots + \beta_n)$ (the sum is commutative), or to $\phi^\alpha(\beta_1 + \cdots + \beta_n + 1)$ when $\phi^\alpha(\beta_1 + \cdots + \beta_n) = \beta_1 + \cdots + \beta_n$ (that is, when $n = 1$ and $\beta_1 = \phi^\gamma(\delta)$, $\alpha < \gamma$).[1] This gives all (the predicative) ordinals up to Γ_0, the first ordinal whose definition requires "things infinite."

One can also consider ordered trees with ordinary nodes, treating its leftmost subtree as the root in the supertree ordering.

See Gallier [1991] for an exposition on properties of these ordinals.

[1] This patches the order-preserving mapping given in [Dershowitz, 1987]. An embedding of trees into Γ_0 is given by [Gallier, 1991] and others. It avoids supernodes, but ignores all subtrees but the two largest.



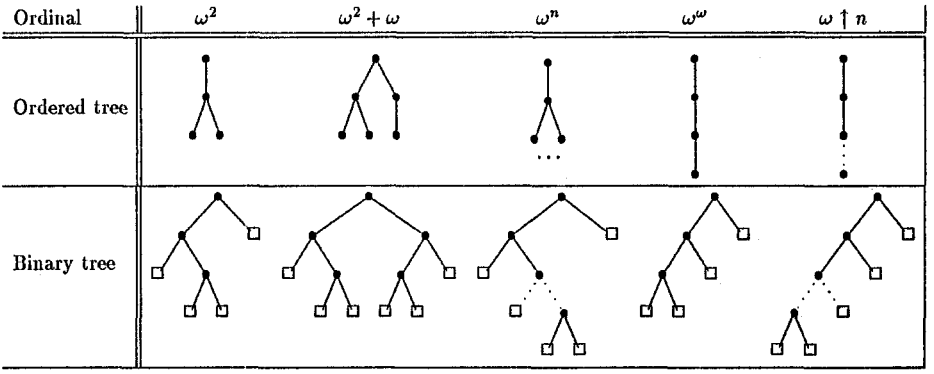


Table 1: Ordinals and trees

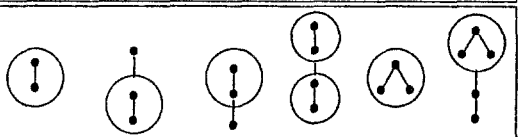
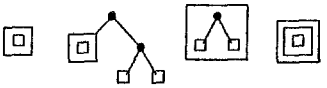
Ordinal	ϵ_0	$\epsilon_0 + 1$	ϵ_1	ϵ_{ϵ_0}	$\phi_2(0)$	$\phi_2(\omega)$
Supertree						
Super leaves					–	–

Table 2: Big ordinals and supertrees

Figure 1: Hercules versus Hydra.

All the orderings described in this section are simplification orderings [Dershowitz, 1982]: a tree is greater than any homeomorphically embedded tree. It has been shown [Okada and Steele, 1988] that all such orderings on finite trees are initial segments of Ackermann's notation (what we got with supertrees), which itself can be proved well-ordered by appealing to Kruskal's Tree Theorem. The well-orderings of rational and infinite trees are consequences of generalizations of the Tree Theorem, but their proof-theoretic strength is unknown.

3 Trees for termination

The contest "Hercules vs. Hydra" was designed [Kirby and Paris, 1982] to be terminating, but not provably so in Peano Arithmetic. Hydra is a bush-like creature with multiple heads. Each time Hercules hacks off a head of hers, Hydra sprouts many new branches identical to the weakened branch that used to hold the severed head, and adjacent to it. If he chops off a head coming straight out of the ground, no new branches result. It cannot be shown by elementary means that Hercules always defeats Hydra, reducing her to nothing, but it can be shown by ϵ_0 induction. See Figure 1.

The appendix contains code for doing arithmetic with ordinals up to ϵ_0, representing them as binary trees in the manner described in the previous section. The nth chop and regrowth steps reduces Hydra's value as an ordinal to its nth predecessor. The code can be used to calculate ordinals used in termination proofs.

Floyd [personal communication] gave the following problem on a Ph.D. qualifying exam at Carnegie-Mellon University in 1967 (expecting the students to use ordinals to solve it): Show that repeatedly applying the rules

$$\begin{array}{rcl} Dt & \to & 1 \\ Da & \to & 0 \\ D(x+y) & \to & Dx + Dy \\ D(x \cdot y) & \to & y \cdot Dx + x \cdot Dy \\ D(x-y) & \to & Dx - Dy \end{array}$$

for symbolic differentiation to an arbitrary expression always ends with a term to which no rule applies. The proof of termination is complicated by the fact that one is allowed to apply a rule at any time to any subexpression of any of the given patterns, rewrite the subexpression accordingly. Termination follows by using the superleaf ordering, viewing expressions as operator trees, but with D as a leaf containing its argument.

The following transformation system was included as an example by Iturriaga [1967] (one of the students solving the above-mentioned "qual" question), but no proof of termination was given therein:

$$\begin{array}{rcl} \neg\neg x & \to & x \\ \neg(x \vee y) & \to & \neg x \wedge \neg y \\ \neg(x \wedge y) & \to & \neg x \vee \neg y \\ x \wedge (y \vee z) & \to & (x \wedge y) \vee (x \wedge z) \\ (y \vee z) \wedge x & \to & (y \wedge x) \vee (z \wedge x) \end{array}$$

To see that it terminates, use supertrees, viewing $\neg$ as 2, $\wedge$ as 1, $\vee$ and constants as 0.

Boyer used the following transformation in his theorem prover and circulated an electronic message (in 1977) soliciting proofs of its termination:

$$if(if(x,y,z),u,v) \quad \to \quad if(x,if(y,u,v),if(z,u,v))$$

This follows directly from Ackermann's original three-place notation. We can view $if(x,y,z)$ as a tree with x as its root and y and z as its two subtrees. The supertree ordering shows termination.

The following set of rewriting rules, an extension of the Hydra contest which allows the tree to grow in height as well as breadth, is designed to mimic Γ_0-induction:

$$\begin{array}{rcl} G_n\overline{x} & \to & G_{n+1}p_nx \\ p_n\langle x,y,z\rangle & \to & \langle x,\overline{y},p_nz\rangle \\ p_{n+1}\langle A,y,z\rangle & \to & \langle A,p_{n+1}y,r_n\langle B,\langle A,y,z\rangle,z\rangle\rangle \\ p_n\langle x,y,z\rangle & \to & \overline{y} \\ p_n\langle B,y,z\rangle & \to & r_n\langle B,y,z\rangle \\ r_{n+1}\langle B,y,z\rangle & \to & \langle B,p_{n+1}y,r_n\langle B,y,z\rangle\rangle \\ r_n\langle x,y,z\rangle & \to & \overline{z} \\ \langle x,\overline{y},\overline{z}\rangle & \to & \overline{\langle x,y,z\rangle} \end{array}$$

"A nodes" are lexicographic; "B nodes" are sums, summands of which r duplicates for p to reduce; G stands for "Gremlin"; the bar keeps track of what p has done. Even bigger battles—and their associated ordinals—are described in [Okada, 1988].

4 Conclusion

We conclude with the function f of Figure 2 (a repaired riddle from [Dershowitz and Reingold, 1992]): Show that the sequence

$$t \quad f(t) \quad f(f(t)) \quad \cdots \quad f^n(t) \quad \cdots$$

always ends in a leaf starting with any finite binary tree t with foliage of two kinds: ♣ and ♠.

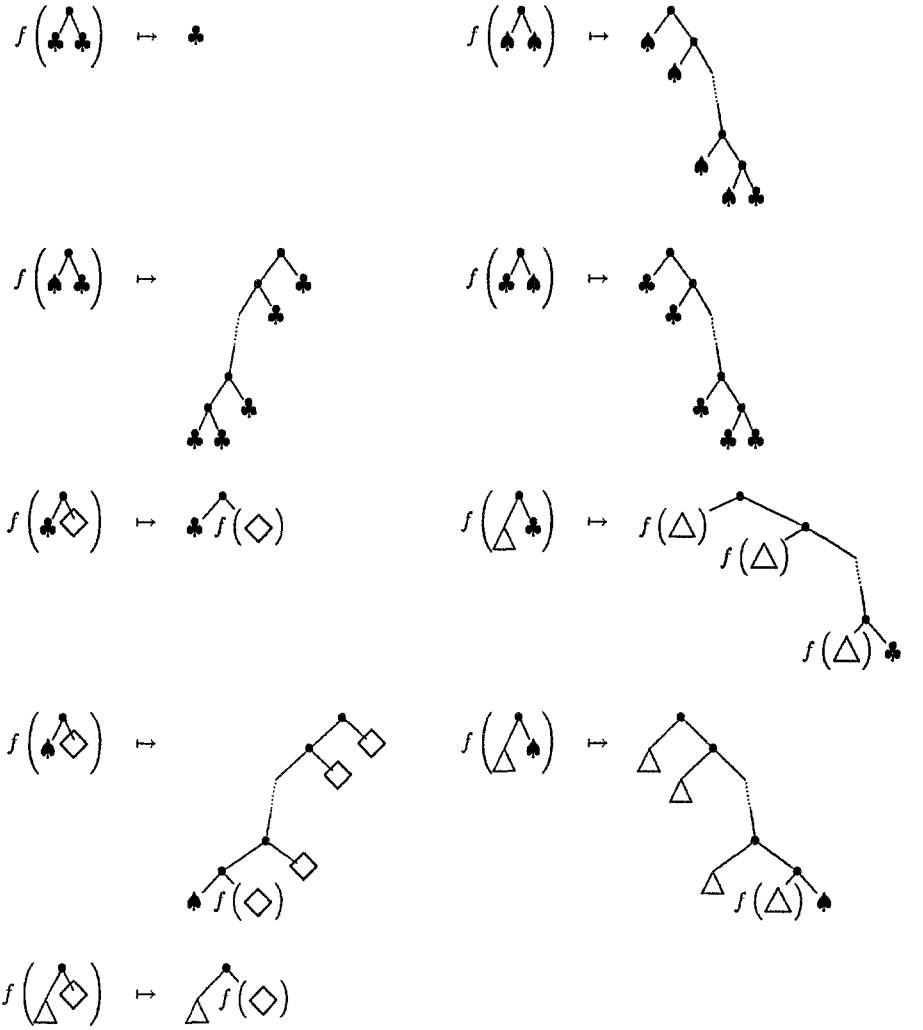


Figure 2: Hybrid tree (triangles and diamonds are arbitrary non-leaf trees)

Note

The LaTeX tree-drawing macros used here (co-authored with E. M. Reingold) are available by anonymous `ftp` on `emr.cs.uiuc.edu` as `/pub/tex/tree.sty`. They require M. J. Wichura's `piclatex.sty`.

Acknowledgement

I thank Mitsu Okada and Ed Reingold for their collaboration on aspects of this work.

References

[Dennis-Jones and Wainer, 1983] E. C. Dennis-Jones and S. S. Wainer. Subrecursive hierarchies via direct limits. In M. M. Richter, *et al.*, editors, *Computation and Proof Theory: Proceedings of the Logic Colloquium, Part II*, pages 117–128, Aachen, 1983. Lecture Notes in Mathematics 1104, Springer-Verlag.

[Dershowitz and Manna, 1979] Nachum Dershowitz and Zohar Manna. Proving termination with multiset orderings. *Communications of the ACM*, 22(8):465–476, August 1979.

[Dershowitz and Reingold, 1992] Nachum Dershowitz and Edward M. Reingold. Ordinal arithmetic with list expressions. In A. Nerode and M. Taitslin, editors, *Proceedings of the Symposium on Logical Foundations of Computer Science*, pages 117–126, Tver, Russia, July 1992. Vol. 620 in *Lecture Notes in Computer Science*, Springer-Verlag, Berlin.

[Dershowitz, 1982] Nachum Dershowitz. Orderings for term-rewriting systems. *Theoretical Computer Science*, 17(3):279–301, March 1982.

[Dershowitz, 1987] Nachum Dershowitz. Termination of rewriting. *J. of Symbolic Computation*, 3(1&2):69–115, February/April 1987. Corrigendum: *4*, 3 (December 1987), 409–410.

[Floyd, 1967] Robert W. Floyd. Assigning meanings to programs. In *Proceedings of Symposia in Applied Mathematics, XIX: Mathematical Aspects of Computer Science*, pages 19–32, Providence, RI, 1967. American Mathematical Society.

[Gallier, 1991] Jean Gallier. What's so special about Kruskal's Theorem and the ordinal Γ_0. A survey of some results in proof theory. *Annals of Pure and Applied Logic*, 53(3):199–260, September 1991.

[Iturriaga, 1967] R. Iturriaga. Contributions to mechanical mathematics. Ph.D. Thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, 1967.

[Kirby and Paris, 1982] Laurie Kirby and Jeff Paris. Accessible independence results for Peano arithmetic. *Bulletin London Mathematical Society*, 14:285–293, 1982.

[Okada and Steele, 1988] Mitsuhiro Okada and Adam Steele. Ordering structures and the Knuth-Bendix completion algorithm. In *Proceedings of the Allerton Conference on Communication, Control, and Computing*, Monticello, IL, 1988.

[Okada, 1988] Mitsuhiro Okada. Note on a proof of the extended Kirby-Paris game on labeled finite trees. *European Journal of Combinatorics*, 9:249–253, 1988.

[Schmidt, 1976] Diana Schmidt. Built-up systems of fundamental sequences and hierarchies of number-theoretic functions. *Arch. Math. Logik*, 18:47–53, 1976.

[Turing, 1950] Alan M. Turing. Checking a large routine. In *Report of a Conference on High Speed Automatic Calculating Machines*, pages 67–69, Institute of Computer Science, University of Toronto, Toronto, Ontario, Canada, January 1950.

Appendix

The following is a Common Lisp implementation of ordinals up to ϵ_0. Included also is the Battle of Hydra and Hercules, played by calling `hydra(h,one)`, where `h` (typically `'(w(w...(w omega)...)))` is the initial Hydra.

```
(defconstant zero                    nil)
(defconstant one                     (list zero))
(defconstant omega                   (list one))

(defun leq? (x y) ; is x less than or equal to y?
  (cond ((equal x zero)              t)
        ((equal y zero)              nil)
        ((equal (car x) (car y))     (leq? (cdr x) (cdr y)))
        (t                           (leq? (car x) (car y)))))

(defun w (x) ; omega to the x
                                     (list x))

(defun cplus (x y) ; commutative sum of x and y
  (cond ((equal x zero)              y)
        ((equal y zero)              x)
        ((leq? (car x) (car y))      (cons (car y) (cplus (cdr y) x)))
        (t                           (cplus y x))))

(defun ctimes (x y) ; commutative product of x and y
  (cond ((equal x zero)              zero)
        ((equal y zero)              zero)
        ((equal (cdr x) zero)        (cons (cplus (car x) (car y))
                                           (ctimes x (cdr y))))
        (t                           (cplus (ctimes (w (car x)) y)
                                            (ctimes (cdr x) y)))))

(defun succ? (x) ; is x a successor?
  (cond ((equal x zero)              nil)
        ((equal (cdr x) zero)        (not (car x)))
        (t                           (succ? (cdr x)))))

(defun pred (x n) ; nth predecessor of limit ordinal x
                  ; predecessor of successor ordinal x
  (cond ((equal x one)               zero)
        ((not (equal (cdr x) zero)) (cons (car x) (pred (cdr x) n)))
        ((succ? (car x))             (ctimes (w (pred (car x) n)) n))
        (t                           (w (pred (car x) n)))))

(defun hydra (x n)
  (if (null x)
      'Hercules-wins
    (hydra (pred x n) (cplus n one))))
```