

A NOTE ON SOME TREE SIMILARITY MEASURES *

Karel CULIK II

Department of Computer Science, University of Waterloo, Waterloo, Ontario, N2L 3G1 Canada

Derick WOOD **

Unit for Computer Science, McMaster University, Hamilton, Ontario, L8S 4K1 Canada

Received 2 March 1982

Keywords: Tree, similarity measure

1. Introduction

Given two structures of the same type, one standard question is: how close are these two structures to each other? One example is the string-to-string correction problem (see [3,7,9] for example), a second is syntax-error repairing in parsers (see [1,5]), and a third is the similarity of two dendrograms [2,6,8]

It is this third example we are concerned with in the present note. We consider labelled and unlabelled trees and search trees of the same size n . We show that two trees of the same type are $O(n)$ and $O(n \log n)$ distance apart, for unlabelled and labelled trees respectively. The basis for the distance measure is the interchange or rotation tree transformation.

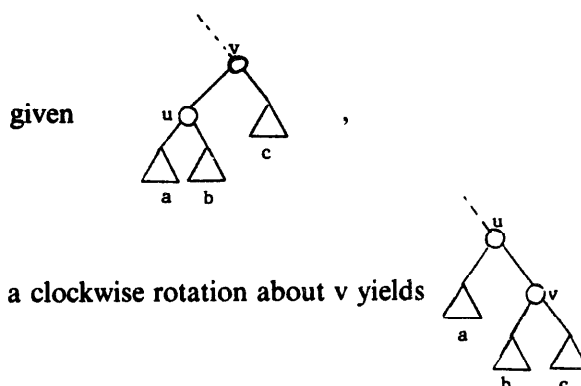
2. The results

We define an (*unrooted*) *binary tree* (or *dendrogram*) to be a connected graph with no cycles, where each node is either unary or ternary. A unary node is called a leaf, external or terminal

node, while a ternary node is called an internal node. If a binary tree has $n > 3$ leaves, then it has $n - 2$ internal nodes and $n - 3$ edges connecting the internal nodes.

Similarly we define a *rooted, oriented and ordered binary tree*, usually called a binary search tree to be a connected digraph, with a designated root node. Each node apart from the root has in-degree 1 and out-degree either 0 or 2. The former are leaves and the latter are internal nodes.

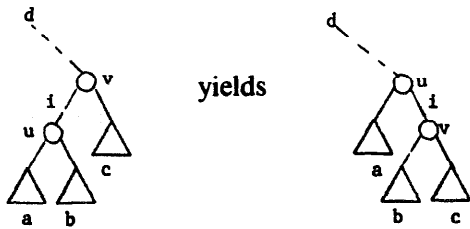
A well known tree transformation in binary search trees is that of rotation or promotion, namely:



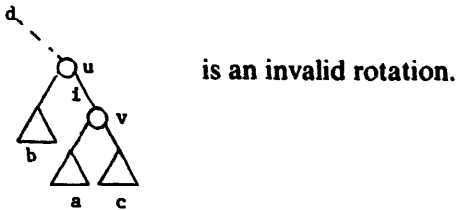
A counterclockwise rotation about u in the second diagram yields the tree in the first diagram. The reasons for the importance of this rotation are two-fold. First, if the original tree is a valid search

* Work carried out under Natural Sciences and Engineering Council of Canada Grant Nos. A-7403 and A-7700.

** Present address: Department of Computer Science, University of Waterloo, Waterloo, Ontario, N2L 3G1 Canada.



that is $\{\{a, b\}, \{c, d\}\}$ yields $\{\{a, d\}, \{b, c\}\}$. This interchange is uniquely defined because the trees are ordered, and therefore



We define the *interchange distance* of S from T , denoted by ' $\text{idist}(S, T)$ ', where S and T are trees with n internal nodes, as we did for rotation distance. Letting F_n denote the class of trees with n nodes, we immediately have the following.

Theorem 2.3. For all $n \geq 1$:

- (i) For all S and T in F_n , $\text{idist}(S, T)$ is well defined and $0 \leq \text{idist}(S, T) \leq 2n - 2$;
- (ii) (F_n, idist) forms a metric space.

However, Waterman and Smith [8] are concerned with *leaf labelled trees*, that is, each leaf of a

given tree T has a unique label associated with it (unique with respect to T , that is). It is convenient, and with no loss of generality, to assume the labels to be the integers $1, 2, \dots$. We use n to denote the number of *leaves* of a tree in this discussion, where $n \geq 3$.

We may once more define a distance measure between labelled trees with n leaves, let us call it the *labelled interchange distance* of S from T , denoted by ' $\text{lidist}(S, T)$ '. However, note that when S and T are isomorphic, this means that they are not only structurally the same, as with idist , but also corresponding leaf nodes have the same label. We now obtain our final theorem, letting L_n denote the class of leaf labelled trees with n leaves.

Theorem 2.4. For all $n \geq 3$:

- (i) For all S and T in L_n , $\text{lidist}(S, T)$ is well defined and

$$0 \leq \text{lidist}(S, T) \leq 4n - 12 + 4n[\log_2(\frac{1}{3}n)];$$

- (ii) (L_n, lidist) forms a metric space.

Proof. (i) $\text{lidist}(S, T)$ is well defined since we can transform S into R which is equal to T , if leaf labels are ignored, via Theorem 2.3. Then for each leaf in R with an incorrect label i , say, it is swapped with the leaf having the required label, j , say (see Fig. 1), and now j can be moved down to its position by the same technique. Clearly at most n such swaps are necessary and it should be observed that the relative ordering of the subtrees on

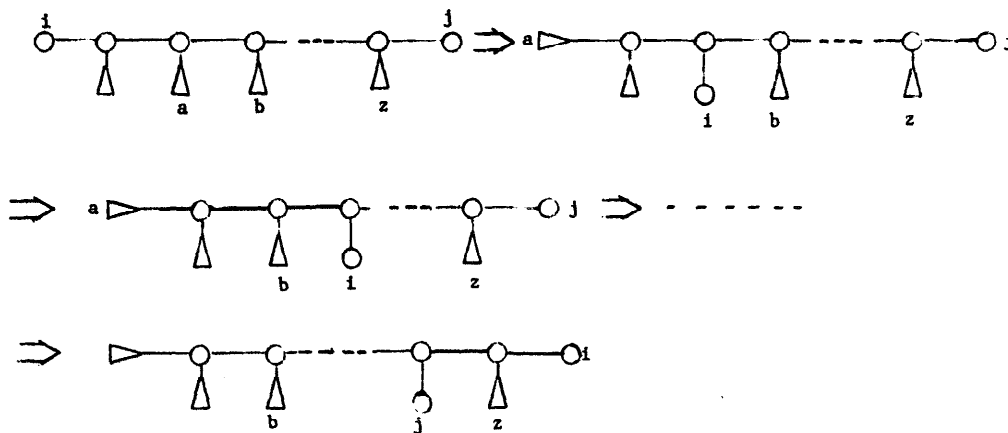
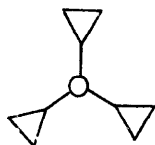


Fig. 1.

the path connecting i and j remains undistributed by the swapping. Hence $\text{lidist}(S, T)$ is well defined.

To show that it is bounded above by $4n - 12 + 4n\lceil\log_2(\frac{1}{3}n)\rceil$, transform S into a minimal diameter tree R , that is, the longest path is minimal. Hence R has the appearance of

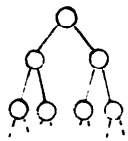


where each subtree has approximately equal height, bounded above by $\lceil\log_2(\frac{1}{3}n)\rceil$. Hence to swap the values i and j at two leaves takes at most $2(2\lceil\log_2(\frac{1}{3}n)\rceil)$ interchanges by the above and n swaps requires at most $4n\lceil\log_2(\frac{1}{3}n)\rceil$ interchanges.

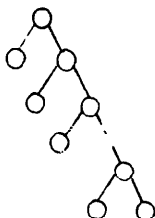
Now, to obtain R , by previous arguments at most $2n - 6$ interchanges are needed and similarly to obtain R from T , yielding the result.

(ii) This follows immediately. \square

As in [8] we leave a number of problems unsolved. For the unlabelled cases we have an $O(n)$ interchange/rotation algorithm and this is clearly asymptotically optimal, since the tree



with n nodes requires $O(n)$ interchanges to give the right spine tree



However for the labelled case we have an $O(n \log n)$ interchange algorithm, but we have no proof of optimality, although we conjecture it to be so. Similarly the concrete bound of Theorem 2.4 is not known to be achievable, but we have no better one. Finally, given two trees S and T , what is the complexity of determining $\text{rdist}(S, T)$, $\text{idist}(S, T)$ or $\text{lidist}(S, T)$?

References

- [1] A.V. Aho and T.G. Peterson, A minimum distance error-correcting parser for context-free languages, *SIAM J. Comput.* 1 (1972) 305-312.
- [2] W.H.E. Day, A new approach to constructing tree metrics, Computer Science Tech. Rept. No. 8001, Memorial University of Newfoundland, 1980.
- [3] D.S. Hirschberg, Complexity of common subsequence problems, Proc. 1977 Fundamentals of Computation Theory Conf., Lecture Notes in Computer Science 56 (Springer, Berlin, 1977) pp. 393-398.
- [4] D.E. Knuth, The Art of Computer Programming, Vol. 3: Sorting and Searching (Addison-Wesley, Reading, MA, 1973).
- [5] G. Lyon, Syntax-direct least-errors analysis for context-free languages: A practical approach. *Comm. ACM* 17 (1974) 3-14.
- [6] G.W. Moore, M. Goodman and J. Barnabas, An iterative approach from the standpoint of the additive hypothesis to the dendrogram problem posed by molecular data sets, *J. Theoret. Biology* 38 (1973) 423-457.
- [7] D. Sankoff, Matching sequences under deletion/insertion constraints, *Proc. Nat. Acad. Sci. USA* 69 (1979) 4-6.
- [8] M.S. Waterman and T.F. Smith, On the similarity of dendrograms, *J. Theoret. Biology* 75 (1978) 789-800.
- [9] R.A. Wagner and M.J. Fischer, The string-to-string correction problem, *J. ACM* 21 (1974) 168-173.