CS-88-33 - ON EFFICIENT ENTREEINGS

ABSTRACT:

A *data encoding* is a formal model of how a logical data structure is mapped into or represented in a physical storage structure. Both structures are complete trees in this paper, and we encode the logical or guest tree in the leaves of the physical or host tree giving a restricted class of encodings called *entreeings*. The *cost* of an entreeing is the total amount that the edges of the guest tree are stretched or dilated when they are replaced by shortest paths in the host tree. We are particularly interested in the *asymptotic average cost* of families of similar entreeings.

Our investigation is a continuation of the study initiated in [6].

AUTHORS: Paul S. Amerins, Ricardo A. Baeza-Yates, Derick Wood

PRICE: $2.00

CS-88-34 - THE SUBSEQUENCE GRAPH OF A TEXT

ABSTRACT:

We define the directed acyclic subsequence graph of a text as the smallest deterministic partial finite automaton that recognizes all possible subsequences of that text. We define the size of the automaton as the size of the transition function and not the number of states. We show that it is possible to build this automaton using $O(n \log n)$ time and space for a text of size n. We extend this construction to the case of multiple strings obtaining a $O(n^2 \log n)$ time and $O(n^2)$ space algorithm, where n is the size of the set of strings. For the later case, we discuss its application to the longest common subsequence problem improving previous solutions.

AUTHOR: Ricardo A. Baeza-Yates

PRICE: $2.00

sent
both
Oct. 27

If you would like to order any reports please forward your order, along with a cheque or international bank draft payable to the **Department of Computer Science, University of Waterloo, Waterloo, Ontario, N2L 3G1, to the Research Report Secretary.**

Please indicate your current mailing address and if you wish to remain on our mailing list.

MAILING ADDRESS: Henry Wolkowicz
C&O Dept

YES, REMAIN ON MAILING LIST

NO, DELETE FROM MAILING LIST

OXFORD UNIVERSITY COMPUTING LABORATORY
PROGRAMMING RESEARCH GROUP

Professor of Computation:
C.A.R. Hoare, FRS

Professor of Numerical Analysis:
K.W. Morton

8–11 Keble Road
Oxford OX1 3QD
Tel: Oxford (0865) 273837 (direct)

October 26, 1988

Research Reports Secretary
Department of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1
CANADA

Dear Sir/Madam

We would like to order one copy of each of the following reports:

CS–88–23 *Pattern matching in trees* by Peng Li. Price $5.00

CS–88–33 *On efficient entreeings* by Paul S. Amerins, Ricardo A. Baeza-Yates, Derick Wood. Price $2.00.

I enclose a cheque for $7.00.

Also enclosed is a list of our current publications for your interest. We are happy to implement exchange arrangements.

Yours sincerely

David Brown

David Brown
Librarian

encl: Cheque for $7.00
List of PRG Technical Monographs to October 1988

sent
Nov. 17/88

Printing Requisition / Graphic Services

15066

1. Please complete unshaded areas on form as applicable.
2. Distribute copies as follows: White and Yellow to Graphic Services. Retain Pink Copies for your records.
3. On completion of order the Yellow copy will be returned with the printed material.
4. Please direct enquiries, quoting requisition number and account number, to extension 3451.

TITLE OR DESCRIPTION: On Efficient Entreeings CS-88-33

DATE REQUISITIONED: August 15/88
DATE REQUIRED: ASAP
ACCOUNT NO.: 1 2 6 6 1 7 6 4 1

REQUISITIONER - PRINT: D. Wood
PHONE: 4456
SIGNING AUTHORITY: S. DeAngelis / D. Wood

MAILING INFO — NAME: Sue DeAngelis
DEPT.: C.S.
BLDG. & ROOM NO.: DC 2314
[X] DELIVER
[ ] PICK-UP

NUMBER OF PAGES: 12
NUMBER OF COPIES: 150

TYPE OF PAPER STOCK: [ ] BOND [ ] NCR ____ PT. [ ] COVER [ ] BRISTOL [ ] SUPPLIED [ ] Alpac Ivory 140M

PAPER SIZE: [ ] 8½ x 11 [ ] 8½ x 14 [ ] 11 x 17 [ ] 10x14 Glosscoat ~~10 pt Rolland Tint~~

PAPER COLOUR: [ ] WHITE [X] ____
INK: [X] BLACK [ ] ____

PRINTING: [ ] 1 SIDE ____ PGS. [X] 2 SIDES ____ PGS.
NUMBERING: FROM ____ TO ____

BINDING/FINISHING: [X] COLLATING [ ] STAPLING [ ] ____ HOLE PUNCHED [ ] PLASTIC RING

FOLDING/PADDING: 7x10 saddle stitched
CUTTING SIZE:

Special Instructions

Beaver Cover

Both cover and inside in black ink please

COPY CENTRE

OPER. NO.	BLDG.	MACH. NO.

DESIGN & PASTE-UP

OPER. NO.	TIME	LABOUR CODE
		D01
		D01
		D01

TYPESETTING

	QUANTITY			
PAP000000				T01
PAP000000				T01
PAP000000				T01

PROOF

PRF				
PRF				
PRF				

NEGATIVES

	QUANTITY	OPER. NO.	TIME	LABOUR CODE
FLM				C01
FLM				C01
FLM				C01
FLM				C01
FLM				C01

PMT

PMT				C01
PMT				C01
PMT				C01

PLATES

PLT				P01
PLT				P01
PLT				P01

STOCK

				001
				001
				001
				001

BINDERY

RNG				B01
RNG				B01
RNG				B01
MIS000000				B01

OUTSIDE SERVICES

$ ____ COST

TAXES — PROVINCIAL [ ] FEDERAL [ ]

GRAPHIC SERV. OCT. 85 482-2

August 12, 1988

Sue:

Can you have 150 copies of each tech.report made up in the "Beaver" cover? Thanks. (126-6176-41)D.Wood

Thanks also for the cgl work - terrific job!

z

UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT

On Efficient Entreeings

Paul S. Amerins,
Ricardo A. Baeza-Yates
and
Derick Wood

Data Structuring Group
Research Report CS-88-33

August, 1988

On Efficient Entreeings *

Paul S. Amerins Ricardo A. Baeza-Yates Derick Wood †

August 1988

Abstract

A *data encoding* is a formal model of how a logical data structure is mapped into or represented in a physical storage structure. Both structures are complete trees in this paper, and we encode the logical or guest tree in the leaves of the physical or host tree giving a restricted class of encodings called *entreeings*. The *cost* of an entreeing is the total amount that the edges of the guest tree are stretched or dilated when they are replaced by shortest paths in the host tree. We are particularly interested in the *asymptotic average cost* of families of similar entreeings.

Our investigation is a continuation of the study initiated in [6]. In particular, the paper contains the following main results.

1. We refute a conjecture in [6] that a particular family of entreeings of binary guests into binary hosts was optimal.
2. We provide an efficient family of entreeings for k-ary guests into k-ary hosts, for $k \geq 2$.
3. We provide an efficient family of entreeings of k-ary guests into binary hosts, for $k \geq 3$.
4. We provide a new simple lower bound technique that can be applied to the entreeings of (2) above to prove that they are very close to optimal. Moreover, it can be adapted for the entreeings of (3) above, in which case we are able to show near optimality when k is sufficiently large.

*The research of the second author was supported by a scholarship from the Institute for Computer Research and that of the third author was supported by an Information Technology Research Centre Grant and by a Natural Sciences and Engineering Research Council of Canada Grant No. A-6192.

†Data Structuring Group, Department of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, CANADA

1 Introduction

A *data encoding* is the physical storage representation of a logical data structure [4]. In this paper, both kinds of structures are represented by complete trees and an encoding is an embedding of a guest tree (the logical structure) into a host tree (the physical structure). Moreover, the nodes of the guest tree are mapped injectively into the leaves of the host tree; this is called an *entreeing* [5,6]. The *cost* of an encoding is the total amount that the edges of the guest tree are dilated when they are replaced by shortest paths in the host tree via the entreeing.

Interest in encodings of trees has recently been revived because of the advent of tree machines; see [1]. In this setting, emulating a tree machine within some other architecture, for example, a hypercube, is modeled by data encoding. Moreover, being able to efficiently entree a logical tree, for example, a divide-and-conquer tree, into a physical tree, that is, a tree machine, is a basic problem.

Throughout, we entree a (complete) guest tree of height h into the leaves of a (complete) host tree of *smallest possible height.* This is reasonable because we are interested in cost efficient entreeings. For, entreeing a guest into a host that is higher than necessary is more expensive unless the entreeing is restricted to a minimal height subtree of the host. Furthermore, we are interested in families of related entreeings, rather than in ad hoc entreeings for particular heights of guests. For this reason, we consider entreeings ϵ_h of a guest of height h that are defined recursively in terms of ϵ_i, $1 \leq i < h$. For each family $\{\epsilon_h : h \geq 1\}$ of *recursive entreeings*, as we call them, we wish to compute its *asymptotic average cost*, for a uniform distribution on the edges of the guest tree. Rosenberg et al. [6] studied, in detail, one such family of recursive entreeings of binary guests into binary hosts and proved that its asymptotic average cost is $5 + o(1)$. They also proved a lower bound of $4.78\dot{6} + o(1)$ for the asymptotic average cost of any family of entreeings, not only recursive ones, and, therefore, conjectured that their family was optimal. Indeed, they said, "Although we have been unable to verify the conjectured optimality of the entreeing ϵ_h, the bound just obtained shows ϵ_h to be close enough to optimal ... to forestall any serious search for a less ACOSTly encoding."

Surprisingly (at least to the third author), a less ACOSTly binary entreeing has been found; it is defined and analyzed in Section 3. Its asymptotic average cost is $4.8\dot{3} + o(1)$; very close indeed to the lower bound $4.78\dot{6} + o(1)$ of [6]. Apart from this refutation, we present efficient entreeings of k-ary guests into k-ary hosts, for $k \geq 3$, in Section 3 and efficient entreeings of k-ary guests into binary hosts, for $k \geq 3$, in Section 5. To complement this extension of the investigation in [6], we provide, in Section 4, a new, simple lower bound technique which can be applied to the entreeing problems we study. This enables us to demonstrate that our new entreeings *are* efficient in that they are close to optimal, for all values of k.

2 Definitions

Let h be the height of a complete k-ary tree, defined as the number of edges in a path from the root to a leaf. The number of nodes of a complete k-ary tree of height h is

$$N_h = \frac{k^{h+1} - 1}{k - 1}$$

and the number of edges is

$$E_h = N_h - 1 = \frac{k^{h+1} - k}{k - 1}$$

and the number of leaves is

$$L_h = k^h$$

Let G_h be a guest tree of height h and H be a host tree. An *entreeing* is an injection of nodes in G_h into the leaves of H. To compute the average cost we use a uniform distribution for the edges of G_h, that is, each edge has the same probability. Thus, we define the *cost* of an entreeing ϵ_h as

$$C_h(\epsilon_h) = \sum_{e \in \ Edges(G)} dilation(\epsilon_h(e))$$

where the dilation of an edge e in the guest tree is the length of the shortest path between the images of the two incident vertices under the entreeing. Given a family $\{\epsilon_h : h \geq 1\}$ of entreeings, the *asymptotic average cost* of an entreeing ϵ_h is defined as

$$A = \lim_{h \to \infty} \frac{C_h(\epsilon_h)}{E_h}$$

We define the *expansion cost* of an entreeing as the size of the host tree divided by the size of the guest tree

$$X = \frac{Nodes(H)}{Nodes(G)}$$

The aim of this paper is to obtain entreeings with minimal expansion and minimal asymptotic average cost.

We define the *inorder traversal* of a k-ary tree recursively as:

> Traverse the first $\lfloor k/2 \rfloor$ left subtrees in inorder; visit the root; and traverse the other subtrees in inorder.

The enumeration induced by this traversal is the enumeration used to represent the vertices of the guest tree.

3 Entreeing k-ary trees into k-ary trees

Let G_h and H be complete k-ary trees ($k \geq 2$). If we are to embed a tree G_h into the leaves of a host tree H, the height of H must be at least $h+1$. Hence, to obtain the minimum expansion, the host tree H must have height exactly $h+1$. This gives an expansion of

$$X = \frac{(k-1)k^{h+1}}{k^{h+1}-1} = k - 1 + O(k^{-h})$$

For simplicity we will consider the case $k = 2$ in detail giving the results for general k later. The simplest entreeing is the *inorder entreeing*; this is the recursive entreeing studied in [6]. We encode the nodes, using the inorder enumeration of G_h, in the leaves of H_{h+1} from left to right; see Figures 1 and 2 for an example. Alternatively and recursively, one can express this as entreeing the left subtree of G_h in the left subtree of H_{h+1} using ϵ_{h-1}, entreeing the right subtree of G_h in the right subtree of H_{h+1} using ϵ_{h-1}, and associating the root of G_h with the rightmost and unused leaf of the left subtree of H_{h+1}. The cost of entreeing a tree of height h can be expressed by the recurrence equation

$$C_h = \begin{cases} 6, & h = 1 \\ 2C_{h-1} + 4h + 2, & h > 1 \end{cases}$$

Its solution is

$$C_h = 5 \cdot 2^{h+1} - 4h - 10$$

which gives an asymptotic average cost of 5 as shown in [6].

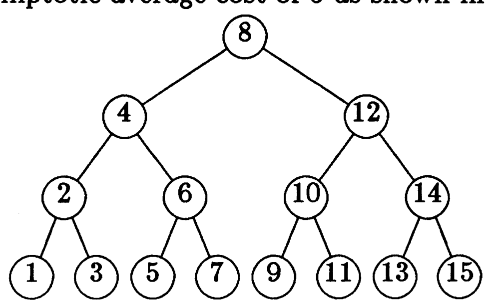

Figure 1: Guest tree

In [6], it was conjectured that this entreeing was optimal, but we now demonstrate that this is not the case. The following algorithm is almost optimal and is also based on the inorder entreeing. If the height is less than 3, the entreeing is identical to the inorder entreeing. Otherwise, we apply the algorithm recursively as depicted in Figure 3. The idea is first to embed, recursively, subtrees g_1, g_2, and g_3 of G_h into the subtrees h_1, h_2, and h_3

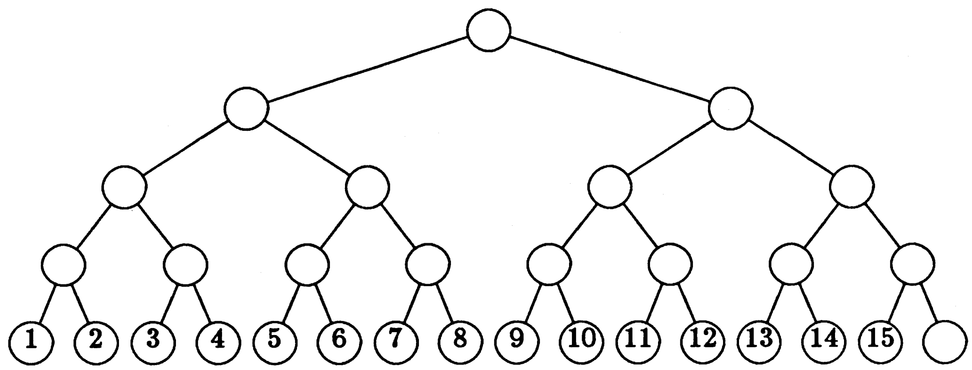

Figure 2: Host tree obtained with the inorder entreeing

of H_{h+1} (after this step, the righmost leaves of h_1, h_2, and h_3 are empty). Second, we modify the partial entreeing ϵ_h constructed so far. Consider the node $\epsilon_h(x)$ in h_2 corresponding to the rightmost leaf x in g_2 and let y be the left child of the root r of G_h. Now, we define $\epsilon_h(r)$ to be the rightmost leaf of h_2, define $\epsilon_h(y)$ to be the current value of $\epsilon_h(x)$, and redefine $\epsilon_h(x)$ to the rightmost leaf of h_1. In the example of Figure 2, the only difference is that nodes 4 and 7 are interchanged.

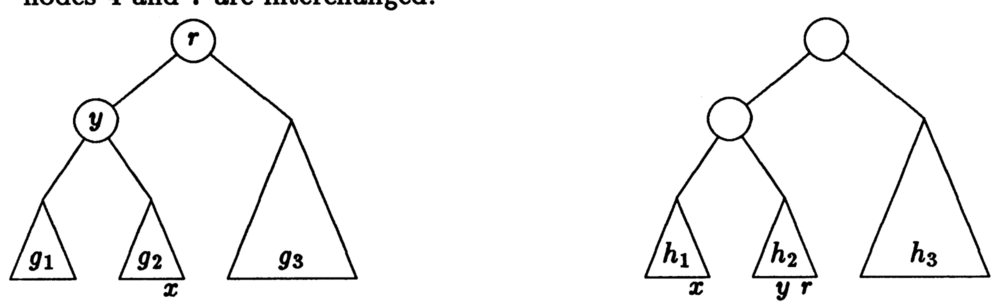

Figure 3: Improved inorder entreeing

Each recursive call increases the dilation of an edge in a lower level, decreasing the dilation of one of the edges at the root to 2, yielding a saving of 2. Note that the leaves of different interchanges do not interfere. We call this entreeing the *improved inorder entreeing*, or *IIE* for short. Its recurrence equation is

$$C_h = \begin{cases} 6, & h = 1 \\ 22, & h = 2 \\ C_{h-1} + 2C_{h-2} + 8h - 2, & h > 2 \end{cases}$$

and its solution is

$$C_h = \frac{29}{6}2^{h+1} + \frac{(-1)^h}{3} - 4h - 9$$

giving an asymptotic average cost of $\frac{29}{6} = 4.8\dot{3}$. This is almost optimal, since a lower bound of 4.786 is given in [6]. Table 1 gives some values for the cost of this entreeing compared with the values for the inorder entreeing for small h.

h	0	1	2	3	4	5
IE	1	6	22	56	134	290
IIE	1	6	22	56	130	280

Table 1: Entreeing values for small h

For general k, the natural way to generalize the inorder entreeing is to apply a divide and conquer strategy. We call this entreeing the *divide-and-conquer-entreeing* or *DCE.* We embed each subtree of G_h into the corresponding subtree of H_{h+1}, mapping the root of G_h to one of the subtrees of H_{h+1}. The corresponding recurrence equation is

$$C_h = \begin{cases} 4k-2, & h=1 \\ kC_{h-1} + 2(k-1)(h+1) + 2h, & h>1 \end{cases}$$

and its solution is

$$C_h = \frac{2(2k^2-2k+1)}{(k-1)^2}(k^h-1) - \frac{2k}{k-1}h$$

giving an asymptotic average cost of

$$A_{DCE} = \frac{2(2k^2-2k+1)}{k(k-1)} = 4 + \frac{2}{k^2} + O(k^{-3})$$

As expected this reduces to 5 for $k=2$. However, it is possible to improve this result considerably. To do this we embed the tree using the generalization of the inorder traversal defined in Section 2. We call this entreeing the *IE entreeing.* The recurrence equation is now

$$C_h = \begin{cases} 2k+2, & h=1 \\ kC_{h-1} + 2(k-1)h + 2(h+1), & h>1 \end{cases}$$

Its solution is

$$C_h = \frac{2(k^2+k-1)}{(k-1)^2}(k^h-1) - \frac{2k}{k-1}h$$

giving an asymptotic average cost of

$$A_{IE} = \frac{2(k^2+k-1)}{k(k-1)} = 2 + \frac{4}{k} + O(k^{-2})$$

To see that this is considerably better, consider the ratio

$$\frac{A_{DCE}}{A_{IE}} = \frac{2k^2 - 2k + 1}{k^2 + k - 1} = 2 - \frac{4}{k} + O(k^{-2})$$

Clearly, A_{DCE} is, asymptotically, almost double the cost of A_{IE}.

Our last entreeing is a generalization of the improved inorder entreeing. It is defined recursively as follows.

1. If the height of G_h is less than 3, use the IE entreeing.
2. Otherwise, for the first $k-1$ subtrees of G_h (g_i) and H_{h+1} (h_i) do
   (a) Recursively entree all the subtrees of g_i (g_i^j) into the corresponding subtrees h_i^j of h_i.
   (b) Map the root of g_i (r) to the free leaf of h_i.
   (c) Interchange $\epsilon_h(r)$ with $\epsilon_h(y)$, where y is the rightmost element of g_i.
3. Entree g_k into h_k.
4. Map the root of G_h to the free leaf of h_k.

The corresponding recurrence equation is

$$C_h = \begin{cases} 2k+2, & h=1 \\ 2k^2+6k+2, & h=2 \\ C_{h-1} + k(k-1)C_{h-2} + 2hk^2 - 2k^2 + 2k + 1, & h>2 \end{cases}$$

and its solution is

$$\begin{aligned} C_h &= \frac{2(2k^4+k^3-4k^2+3k-1)}{k^2(2k^2-3k+1)(k-1)}k^{h+1} \\ &- \frac{2}{k(2k^2-3k+1)}(1-k)^{h+1} - \frac{2k}{k-1}h - \frac{2(k^3+k-1)}{k(k-1)^2} \end{aligned}$$

yielding an asymptotic average cost of

$$A_{IIE} = \frac{2(2k^4+k^3-4k^2+3k-1)}{k^2(2k^2-3k+1)} = 2 + \frac{4}{k} + O(k^{-2})$$

We can compare A_{IE} and A_{IIE} by examining the ratio

$$\frac{A_{IE}}{A_{IIE}} = 1 + \frac{1}{2k^2} + O(k^{-3})$$

In this case the difference is slight, since both equal two in the limit.

Table 2 gives some values for the asymptotic average cost of the three previous entreeings for some values of k along with the known lower bounds.

k	Lower Bound	IIE	IE	DCE
2	4.78 [6]	4.83	5	5
3	3.15	3.58	3.67	4.33
4	2.89	3.11	3.16	4.17
5	2.73	2.86	2.90	4.10
6	2.62	2.71	2.73	4.07
7	2.53	2.60	2.62	4.05
8	2.47	2.52	2.54	4.04
9	2.42	2.46	2.47	4.03
.	.	.	.	.
∞	2	2	2	4

Table 2: Asymptotic average cost for different k

4 A New Lower Bound Technique

Rosenberg *et al* [6] gave an involved packing argument that led to the lower bound for binary-binary entreeings. We now give a simpler approach that works for general k. For this purpose, we decompose the entreeing problem into two subproblems. We separately consider the guest subtrees of height j (subtrees on the bottom) and the remainder of the guest tree (the top) that we call T. Now, the cost C of any entreeing is bounded below by

$$C \geq \mathit{Dilation}(\mathit{subtrees\ of\ height\ } j) \ + \ \mathit{Dilation}(T)$$

Let C_j be the cost of an optimal entreeing of a tree of height j. A tree of height h has k^{h-j} subtrees of height j. Each internal node in T, apart from the root, is incident to $k+1$ edges. Hence, each internal node in T, except the root, contributes at least $2k+6$ to the cost of any entreeing. This is simply because at most $k-1$ nodes can be at distance two from any other node. Substituting these values into the above inequality we obtain

$$A \geq \frac{(k-1)C_j}{k^{j+1}} + \frac{2k+6}{(k+1)k^j}$$

The cost of an optimal entreeing of a tree of height 1 is $2k+2$. This is because the root can be at distance 2 from at most $k-1$ children. The remaining child must be at distance four, at least. A similar argument shows that C_2 is at least $2k^2+6k+2$. Substituting this value into the previous equation, for $j=2$, we obtain

$$A_k \geq \frac{2(k^4+3k^3+k^2-1)}{(k+1)k^3}$$

Table 2 gives the values of this lower bound for different values of k. An exhaustive search of all possible entreeings for binary trees has proved that the IIE entreeing is optimal for $h = 3$.

5 Entreeing k-Ary Trees into Binary Trees

Since a host tree represents the physical structure, it is more usual for it to be binary. Hence, we would like to entree a k-ary tree ($k > 2$) into a binary tree. To entree a k-ary tree of height h we need a binary tree of at least height h', where

$$2^{h'} \geq \frac{k^{h+1}-1}{k-1}$$

giving $h' \approx h \log_2 k$. To simplify the relation we will use a binary tree of height $h \times b$ with $b = \lceil \log_2(k+1) \rceil$. This gives an expansion factor of

$$\frac{(k-1)2^{hb}}{k^{h+1}-1}$$

which is no larger than

$$\frac{k-1}{k}\left(\frac{k+1}{k}\right)^h$$

We define the inorder entreeing by entreeing each group of k consecutive leaves of the guest tree as the leaves of a subtree of height b in the host tree H. This leaves one free leaf for every k leaves in the host tree. Now, we apply this algorithm recursively (bottom-up) treating the next level of G as leaves and the level of the subtrees of height b as leaves in H.

The recurrence equation for this entreeing is (k is a power of two minus one for simplicity)

$$C_h = \begin{cases} (b-1)2^{b+1} + 2, & h = 1 \\ kC_{h-1} + 2(hb-1)(2^b - 1) + 2b, & h > 1 \end{cases}$$

Its solution is

$$C_h = \frac{2^{b+1}(k(b-1)+1) + 2(k-b-1)}{(k-1)^2}(k^h - 1) - \frac{2b(2^b-1)}{k-1}h$$

giving an asymptotic average cost of

$$A_{IE} = \frac{2^{b+1}(k(b-1)+1) + 2(k-b-1)}{k(k-1)} = 2\log_2 k + O(1)$$

Using a similar lower bound technique to that of the previous section with $j = 1$ we have

$$A \geq \frac{(k-1)C_1}{k^2} + \frac{2}{k}$$

because each edge in T has at least a dilation of 2. We use the value of C_1 given by the inorder entreeing because it is optimal for this height. Thus

$$A \geq \frac{(k-1)((b-1)2^{b+1}+2)+2k}{k^2} = 2\log_2 k + O(1)$$

Hence, the inorder entreeing is asymptotically optimal in k. Table 3 shows some values for the lower bound and the inorder entreeing.

k	Lower Bound	IE
3	2.89	5.33
7	4.45	5.86
15	6.23	7.10
31	8.12	8.66
63	10.06	10.38

Table 3: Asymptotic average cost of entreeing k-ary trees into binary trees

6 Concluding Remarks

We first comment on the entreeing of other tree-like structures and second we state some open problems.

We consider four variations of binary trees that include additional edges. For example, a complete binary tree with edges added between two consecutive leaves if they have different parents. In this case, the asymptotic average cost of the inorder entreeing is 3.25. That is, adding one edge for every two leaves, we decrease the asymptotic average cost by 35%. Now, if we add one edge between every pair of consecutive leaves, the asymptotic average cost decreases to 2.

An interesting tree-like structure is a *dree*. Drees were introduced in [6]. A dree is a tree with undirected threads added such that there is a path of length 2 between adjacent leaves passing through the root of the smallest subtree containing both leaves. These trees are also called *depth inorder trees* [2].

It is shown in [6] that the dilation of any edge in any *endreeing*, is no more than 3, and that the asymptotic average cost for the inorder endreeing is 2.5. Using a similar lower bound argument to that in Section 4 we can prove that this endreeing is optimal.

Another interesting tree-like structure is *depth preorder trees*. Depth preorder trees were introduced in [2]. In these trees, each leaf has an edge to the next node in the preorder traversal. In this case, the inorder entreeing has an average cost of 2.5. A lower bound for this case, using a similar method, is 1.75.

Finally, we have still been unable to prove the optimality of the improved inorder entreeing for binary guests to binary hosts. However, our new upper bound is within .05 of optimal. It would be useful to extend our new lower bound technique to larger subtrees so that the lower bound of [6] can be improved. This implies that we need to find an optimal entreeing of height 4 trees and this is too time consuming for naive enumeration.

Also the lower and upper bounds for k-ary guests to binary hosts are rather far apart — only getting close for $k \geq 31$.

Historical Remark

The third author ran a problem-solving seminar for new Ph.D. students in the 1986 Fall term at the University of Waterloo. One of the problems discussed was the conjecture of [6]. A number of students found the counterexample described here as the improved inorder entreeing, but only the second author of this paper provided asymptotic average cost for general k. It was decided to write this up with P.S. Amerins as the first author, since this is a weak anagram of P. S. Seminar which denotes the nine students in the course.

References

[1] Bhatt, S., Chung, F., Leighton, F.T., and Rosenberg, A.L. "Optimal Simulations of Tree Machines", *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, Toronto, 1986, 274-282.

[2] Jia-Wei, H. and Rosenberg, A. "Graphs that are almost Binary Trees", *Proceedings STOC*, Milwaukee, 1981, 334-341.

[3] Jia-Wei, H., Mehlhorn, K. and Rosenberg, A. "Cost Trade-offs in Graph Embeddings, with Applications", *Journal of the ACM* **30**, 709-728 (1983).

[4] Rosenberg, A.L. "Data Encodings and Their Costs", *Acta Informatica* **9**, 273-292 (1978).

[5] Rosenberg, A.L. "Encoding Data Structures in Trees", *Journal of the ACM* **26**, 668-689 (1979).

[6] Rosenberg, A.L., Wood, D., and Galil, Z. "Storage Representations for Tree-Like Data Structures", *Mathematical Systems Theory* **13**, 105-130 (1979).