The construction of Huffman codes is a submodular ('convex') optimization problem over a lattice of binary trees

D. Stott Parker
stott@cs.ucla.edu
Computer Science Department
University of California
Los Angeles, CA 90095-1596

Prasad Ram
prasad@cp10.es.xerox.com
Xerox Corporation
El Segundo, CA 90245

October 18, 1996

Abstract

We show that the space of all binary Huffman codes for a finite alphabet defines a *lattice*, ordered by the imbalance of the code trees. Representing code trees as path-length sequences, we show that the imbalance ordering is closely related to a majorization ordering on real-valued sequences that correspond to discrete probability density functions. Furthermore, this tree imbalance is a partial ordering that is consistent with the total orderings given by either the external path length (sum of tree path lengths), or the entropy determined by the tree structure. On the imbalance lattice, we show the weighted path-length of a tree (the usual objective function for Huffman coding) is a *submodular* function, as is the corresponding function on the majorization lattice. Submodular functions are discrete analogues of convex functions. These results give perspective on Huffman coding, and suggest new approaches to coding as optimization over a lattice.

1 Introduction

Traditionally, the Huffman algorithm has be relied upon for developing optimal binary codes. Although the algorithm is quite elegant, it is tricky to prove correct and to reason about. While there is little hope of improving on the complexity of the Huffman algorithm itself[1], there is still room for improvement in our understanding of the algorithm.

It is important to reconsider the foundations of Huffman coding. Although it is remarkably robust in general and still finds widespread use, it is far from optimal in many real applications. The situation has lead to a rapidly growing literature, with diverse theoretical and practical contributions. Huffman coding is optimal only when the symbols to be coded occur independently with certain probabilities. Time-varying dependencies are not captured by the Huffman coding model, and optimal encoding of finite messages is not captured either.

[1]The algorithm is essentially equivalent to sorting, in the sense that the sorted sequence of a sequence of integer values ($x_1 \cdots x_n$) is obtainable directly from the optimal code tree for the values ($2^{x_1} \cdots 2^{x_n}$) (e.g., [15, p.335]).

Our motivation came from analysis of dynamic Huffman coding, a specific extension of Huffman coding in which the code used evolves over time. Recently dynamic coding algorithms have been studied heavily. Our initial idea was to define 'rebalancing' operations on code trees and to use these dynamically ('on the fly') in producing better codes, in situations where the distribution of symbols to be coded varies over time and/or is not accurately predictable in advance.

In a sense, this paper reconstructs Huffman coding, viewing it as an optimization over the space of binary trees and seeking a natural representation for this space. Trees are represented as sequences of ascending path-lengths, since that is what is significant in producing optimal codes.

We show in this paper that the set of path-length sequences representing binary trees forms a lattice, which we call the **imbalance lattice**. This lattice orders trees by their imbalance, and gives an organization for them that is useful in optimization. Our belief is that the better mathematical (and not purely procedural) understanding of coding this provides will ultimately pay off in improved algorithms.

The imbalance lattice and its imbalance ordering on trees depend on **majorization** in an essential way. Majorization is an important ordering on sequences that has many applications in pure and applied mathematics [16], which we have related directly to greedy algorithms [21]. Earlier majorization was recognized as an important property of the internal node weights produced by the Huffman algorithm [7, 20], and in this work we go further to clarify its pervasive role.

By viewing the space of trees as a lattice, a variety of new theorems and algorithms become possible. For example, the objective functions commonly used in evaluating codes are **submodular** on this lattice. Submodular functions are often easy to optimize [1, 12, 14, 13, 3], and were shown by Lovász to be closely related to convex functions [14]. Huffman coding gives an important new example of the significance of submodularity in basic algorithms.

2 Ordered Sequences, Rooted Binary Trees, and Huffman Codes

2.1 Ordered Sequences

By a **sequence** we mean an ordered collection of positive real values

$$\mathbf{x} \quad = \quad (\ x_1\ x_2\ \cdots\ x_n\ ).$$

The **length** of this sequence is n. Repetition of values in the sequence is permitted; the values x_j need not be distinct.

We introduce several useful operators on sequences:

ascending sort	$\mathrm{sort}{\uparrow}[\,\mathbf{x}\,]$	$=$	$\langle$ $\mathbf{x}$ put in ascending order $\rangle$
descending sort	$\mathrm{sort}{\downarrow}[\,\mathbf{x}\,]$	$=$	$\langle$ $\mathbf{x}$ put in descending order $\rangle$
sequence exponential	$2^{-\mathbf{x}}$	$=$	$(\ 2^{-x_1}\ \cdots\ 2^{-x_n}\ )$
sequence logarithm	$-\log_2(\mathbf{x})$	$=$	$(\ -\log_2(x_1)\ \cdots\ -\log_2(x_n)\ )$.

A **density sequence** is a nonnegative real-valued sequence whose entries total to 1.

A **distribution sequence** is an ascending nonnegative real sequence whose final entry is 1.

For simplicity, throughout this paper **many sequences are implicitly sorted**, as follows:

$\boldsymbol{\ell}$, $\mathbf{s}$, $\mathbf{t}$	denote **ascending** sequences of positive **integer values** whose sequence exponentials $2^{-\boldsymbol{\ell}}$, $2^{-\mathbf{s}}$, $2^{-\mathbf{t}}$ are density sequences.
$\mathbf{u}$, $\mathbf{v}$	denote **ascending** sequences of positive **real values** that are also density sequences.
$\mathbf{w}$	denotes a **descending** sequence of positive **real values**.
$\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$	denote **descending** sequences of positive **real values** that are also density sequences.

Note since $\boldsymbol{\ell}$ is ascending, $2^{-\boldsymbol{\ell}}$ is descending; and since $\mathbf{x}$ is descending, $-\log_2(\mathbf{x})$ is ascending.

We also allow **sequences** to be operated upon as **vectors**. Thus, if $\mathbf{x}$ is a sequence (vector) of length n, and A is an $n \times n$ matrix, then $A\,\mathbf{x}$ is a sequence (vector). Treating sequences as vectors allows us to define several useful operators with matrix algebra.

2.2 Rooted Binary Trees and Path-Length Sequences

Rooted binary trees here are binary trees with a root node, in which every node is either a leaf node, or an internal node having one parent and two children. Here the order of the leaves is not significant, so a given tree is determined (up to permutation of the leaves) by the lengths of the paths from the root node to each leaf node (the distance of the leaf from the root). Thus we can represent equivalence classes of the rooted binary trees with n leaves by sequences of n nonnegative integers, which give the path-length of each leaf. For example, the path-length sequence

$$(\ 1 \ 3 \ 3 \ 4 \ 4 \ 4 \ 4 \ )$$

represents a binary tree with $n = 7$ leaves, of which one has path-length 1, two have path-length 3, and four have path-length 4; it is shown in Figure 1.

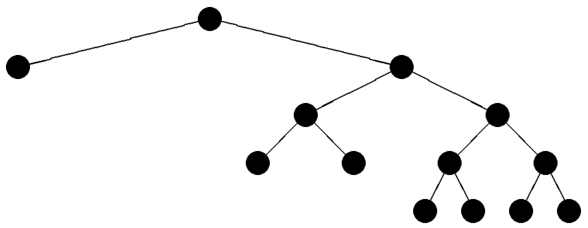

Figure 1: A binary tree having path-length sequence (1 3 3 4 4 4 4)

Path-length sequences obey what we call the **Kraft equality**, a special case of the Kraft inequality of noiseless coding theory (see e.g., [4], p.45).

Theorem 1 *For all $n \geq 1$, $(\ \ell_1 \ \cdots \ \ell_n \ )$ is the sequence of path-lengths in a rooted binary tree* **iff**

$$\sum_{i=1}^{n} 2^{-\ell_i} \quad = \quad 1.$$

Thus $\boldsymbol{\ell}$ is a path-length sequence **iff** $2^{-\boldsymbol{\ell}}$ *is a density sequence.*

Proof
Easily proven by induction on n. For the basis, with $n = 1$ we must have $\ell_1 = 0$. The induction step follows by noticing that the two principal subtrees of any binary tree must have sequences $(\ell'_1 \ \cdots \ \ell'_p)$ and $(\ell''_1 \ \cdots \ \ell''_q)$, each of which satisfies the equality, and their composition has the sequence $(\ (\ell'_1+1) \ \cdots \ (\ell'_p+1) \ (\ell''_1+1) \ \cdots \ (\ell''_q+1) \ )$, which again satisfies the equality. $\square$

Henceforth we assume that **tree path-length sequences are in ascending sorted order**. Below is a lexicographic tabulation of all possible sequences for $1 \leq n \leq 7$, along with T_n, the total number of inequivalent sequences of length n:

n	1	2	3	4	5	6	7	8	9	10	11	12
T_n	1	1	1	2	3	5	9	16	28	50	89	159
	(0)	(1 1)	(1 2 2)	(1 2 3 3) (2 2 2 2)	(1 2 3 4 4) (1 3 3 3 3) (2 2 2 3 3)	(1 2 3 4 5 5) (1 2 4 4 4 4) (1 3 3 3 4 4) (2 2 2 3 4 4) (2 2 3 3 3 3)	(1 2 3 4 5 6 6) (1 2 3 5 5 5 5) (1 2 4 4 4 5 5) (1 3 3 3 4 5 5) (1 3 3 4 4 4 4) (2 2 2 3 4 5 5) (2 2 2 4 4 4 4) (2 2 3 3 3 4 4) (2 3 3 3 3 3 3)	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

An upper bound on T_n can be obtained from the Catalan number C_n, which computes the number of unordered binary trees: for $n \geq 3$, $T_n \leq \frac{1}{2} C_n \leq 2^{n-3}$. Gilbert [6], using the notation $g(N)$ for T_N, points out that T_n is well approximated by

$$T_n \quad \simeq \quad 0.148\,(1.791)^n$$

for $n \leq 30$.

2.3 Huffman Codes are Optimal Path-Length Sequences

In Huffman coding, we are given positive weights

$$w_1 \;\geq\; w_2 \;\geq\; \cdots \;\geq\; w_n$$

and are asked to find a binary tree path-length sequence $\boldsymbol{\ell} = (\,\ell_1\,\ell_2\,\cdots\,\ell_n\,)$ (whose corresponding tree determines a code) such that the weighted path-length

$$g_{\mathbf{w}}(\boldsymbol{\ell}) \quad = \quad \sum_{i=1}^{n} w_i\,\ell_i$$

is minimal. It is evident that we want the path-length sequence to be in ascending sorted order

$$\ell_1 \;\leq\; \ell_2 \;\leq\; \cdots \;\leq\; \ell_n$$

but beyond this and the Kraft equality of Theorem 1, it is difficult to characterize $\boldsymbol{\ell}$.

Huffman's breakthrough [9] was to identify an efficient algorithm that finds an optimal tree, avoiding a search over the exponentially large space of trees. The algorithm repeatedly combines the two tree leaves with least weight, whose sum becomes the weight of a new leaf. Using the algorithm, for example, we can find the optimal tree for the weight sequence $\mathbf{w} = (\ 10\ 9\ 8\ 7\ 3\ 2\ 1\ )$ with $n = 7$. The Huffman (optimal) tree for this sequence has path lengths $\boldsymbol{\ell} = (\ 2\ 2\ 2\ 3\ 4\ 5\ 5\ )$ with total weighted path-length 102:

Path-length sequence $\boldsymbol{\ell}$	$g_{\mathbf{w}}(\boldsymbol{\ell})$
(1 2 3 4 5 6 6)	113
(1 2 3 5 5 5 5)	117
(1 2 4 4 4 5 5)	115
(1 3 3 3 4 5 5)	109
(1 3 3 4 4 4 4)	113
(2 2 2 3 4 5 5)	102
(2 2 2 4 4 4 4)	106
(2 2 3 3 3 4 4)	104
(2 3 3 3 3 3 3)	110

The Huffman algorithm reflects a divide-and-conquer structure that has interesting properties on the space of trees, but because of its procedural nature does little to characterize optimal trees.

3 The Imbalance Lattice of Binary Trees

The optimality of a Huffman code is determined by the match between the balance (or imbalance) between the code tree and the weights of the symbols to be coded. In this section we show ternary balancing exchanges give an imbalance ordering on binary trees that defines a lattice.

We have not found this lattice characterization of tree imbalance elsewhere. Following considerable work in the early 1980's on enumeration of trees, Pallo classified trees by their *rotational* structure (e.g., [18, 19]), and showed that they then form a lattice. However, our work differs from Pallo's in that we classify trees by their *path-length* (imbalance) structure.

3.1 Important Properties of Tree Path-Length Sequences

Theorem 2 *Every path-length sequence* $\boldsymbol{\ell}$ *has the form*

$$\boldsymbol{\ell} \quad = \quad (\;\; \cdots \;\; (q-j) \;\; \overbrace{q \quad \cdots \quad q}^{2k} \;\;)$$

a sequence including $2k$ *copies of its largest value* q *(where* $j,\, k > 0$*). Also, when* k *is a power of 2, it is necessary that* $j = \log_2(2k)$*.*

Proof
Otherwise $\left(2^q \cdot \sum_{i=1}^n 2^{-\ell_i}\right)$ is odd, contradicting the Kraft equality. When k is a power of 2, using this argument again on the shorter path-length sequence obtained by replacing the $2k$ copies of q with one value $(q-\log_2(2k))$, the Kraft equality requires not only $j > 0$ but also $j = \log_2(2k)$. □

Theorem 3 *Except for the sequence* $(\ 1\ 2\ 3\ \ldots\ (n-2)\ (n-1)\ (n-1)\ )$*, any path-length sequence contains at least three identical values.*

Proof
By induction on the length n of the sequence. For the basis, when $n = 3$ the only sequence is $(\ 1\ 2\ 2\ )$, satisfying the theorem. For the induction step, suppose $n > 3$, and to the contrary of the theorem that there is a sequence does not have three identical values. Let q be the smallest value in the sequence appearing twice. We may assume $q < (n-1)$, since otherwise the sequence is $(\ 1\ 2\ 3\ \ldots\ (n-2)\ (n-1)\ (n-1)\ )$. Construct the sequence of length $n-1$ that results from replacing the two values q with one value $(q-1)$. In this new sequence, q does not appear at all (since there were only two before), and $(q-1)$ appears at most twice. Therefore, by induction, since this sequence does not have three identical values it is $(\ 1\ 2\ 3\ \ldots\ (n-3)\ (n-2)\ (n-2)\ )$. But since $q < (n-1)$ and q does not appear in the new sequence, this gives a contradiction. □

3.2 Ternary Exchanges determine Tree Imbalance

The insight that inspired us to write this paper is that **it is possible to generate all binary tree path-length sequences using ternary exchanges**. Given any path-length sequence

$$(\cdots \qquad p \qquad \cdots\cdots\cdots \quad (q+1) \quad (q+1) \quad \cdots)$$

then the revision

$$(\cdots \quad (p+1) \quad (p+1) \quad \cdots\cdots\cdots \qquad q \qquad \cdots)$$

is also a path-length sequence, because

$$2^{-p} + 2^{-(q+1)} + 2^{-(q+1)} \quad = \quad 2^{-p} + 2^{-q} \quad = \quad 2^{-(p+1)} + 2^{-(p+1)} + 2^{-q}.$$

Moreover, if the initial sequence is sorted in ascending order and $p < q$, then the resulting sequence is still a sorted sequence. (When $p = q$ the two sequences are identical.) Dually, this exchange can be applied in reverse; with sorted sequences we can replace the leftmost two $(p+1)$s with one p, and the rightmost $(q-1)$ with two qs, and the result will still be sorted in ascending order.

The net effect of this exchange is to transfer two leaves dangling from level q to level p. The two examples in Figure 2 show this pictorially.

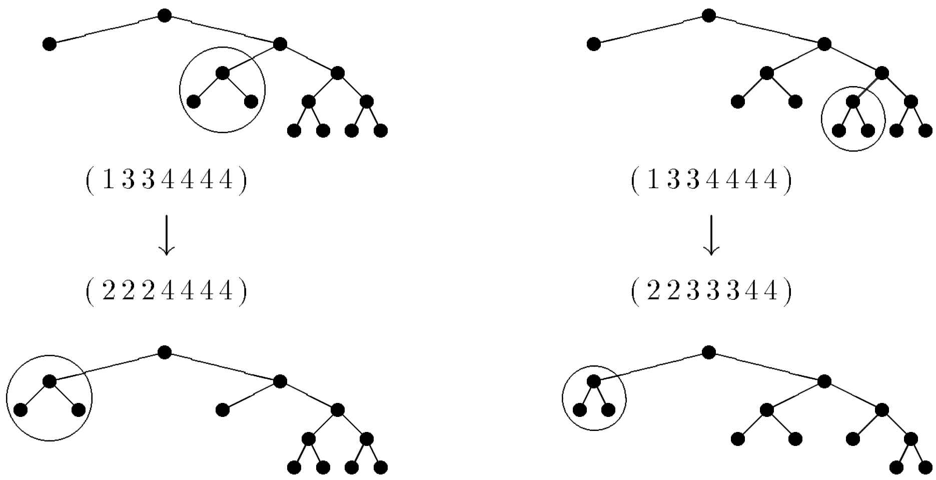

Figure 2: Balancing exchanges: (1 3 3 4 4 4 4) → (2 2 2 4 4 4 4) and (1 3 3 4 4 4 4) → (2 2 3 3 3 4 4)

Definition 1 *Let p, q be integers such that $1 \leq p < q < n$.*
A **balancing exchange** *is a ternary exchange of the form*

$$\begin{array}{cccccccc}
(\cdots & p & & \cdots\cdots & & (q+1) & (q+1) & \cdots) \\
 & \downarrow & \searrow & & \ddots & & \searrow\ \downarrow & \\
(\cdots & (p+1) & (p+1) & & \cdots\cdots & & q & \cdots)
\end{array}$$

It is called a **minimal balancing exchange** *if $(p+1) = q$.*
An **imbalancing exchange** *is of the reverse form*

$$\begin{array}{cccccccc}
(\cdots & (p+1) & (p+1) & & \cdots\cdots & & q & \cdots) \\
 & \downarrow & \swarrow & & \cdots & & \swarrow\ \downarrow & \\
(\cdots & p & & \cdots\cdots & & (q+1) & (q+1) & \cdots)
\end{array}$$

Finally, we can define partial orders as the reflexive transitive closures of these relations among sequences. Given two sequences $\mathbf{s}$ *and* $\mathbf{t}$*, we say* **s is at least as balanced as t***, written*

$$\mathbf{s} \quad \trianglelefteq \quad \mathbf{t}$$

if there are sequences $\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_m$ $(m \geq 1)$ where $\mathbf{t} = \boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_m = \mathbf{s}$ and for each i, $1 \leq i < m$, there is a balancing exchange from $\boldsymbol{\ell}_i$ to $\boldsymbol{\ell}_{i+1}$.

Minimal balancing exchanges, in which $(p+1) = q$, are particularly significant. The balancing exchange (1 3 3 4 4 4 4) → (2 2 2 4 4 4 4) in Figure 2 gives an example, and the center column in Figure 3 gives four examples. Minimal balancing exchanges are *ternary exchanges of consecutive length values*, so that any tree path-length sequence of the form $(\cdots\ (q-1)\ \cdots\ (q+1)\ (q+1)\ \cdots)$ determines the more balanced tree path-length sequence $(\cdots\ q\ \cdots\ q\ q\ \cdots)$ and vice-versa.

Theorem 4 *If two path-length sequences differ, they differ in at least three values. Moreover, if they differ in exactly three values, there is a ternary exchange between the two sequences.*

Proof
Direct consequence of the Kraft equality. The equality shows two path-length sequences cannot differ in one value. Similarly, there cannot be sequences **s** and **t** differing in two values, since if the differences were the disjoint sequences of positive integers ($s_i\ s_j$) and ($t_i\ t_j$) then the Kraft equality would imply $2^{-s_i} + 2^{-s_j} = 2^{-t_i} + 2^{-t_j}$, which is false under the disjointness condition. Similarly sequences differing in three integer values ($s_i\ s_j\ s_k$) and ($t_i\ t_j\ t_k$) must satisfy $2^{-s_i} + 2^{-s_j} + 2^{-s_k} = 2^{-t_i} + 2^{-t_j} + 2^{-t_k}$, and a case analysis shows that this is solved only by ($s_i\ s_j\ s_k$) = ($p\ (q{+}1)\ (q{+}1)$) and ($t_i\ t_j\ t_k$) = ($(p{+}1)\ (p{+}1)\ q$), corresponding to a ternary exchange. □

Ternary exchange transformations can be used to derive any tree path-length sequence from an initial sequence. For example, all binary tree sequences for $n = 6$ can be derived from the (least balanced) initial sequence (1 2 3 4 5 5) using balancing exchanges. The sequences and their derivations are shown pictorially in Figure 3; the center derivation sequence uses only minimal balancing exchanges, and derives each of the five possible path-length sequences for $n = 6$. The right derivation sequence uses a non-minimal balancing exchange.

$$
\begin{array}{cccccccc}
(& 1 & 2 & 3 & 4 & 5 & 5 &) \\
 & \downarrow\searrow & \ddots & \ddots & \ddots & \searrow\downarrow & & \\
(& 2 & 2 & 2 & 3 & 4 & 4 &) \\
 \\
(& 1 & 2 & 4 & 4 & 4 & 4 &) \\
 & \downarrow\searrow & \ddots & \searrow\downarrow & & & & \\
(& 2 & 2 & 2 & 3 & 4 & 4 &)
\end{array}
\qquad
\begin{array}{cccccccc}
(& 1 & 2 & 3 & 4 & 5 & 5 &) \\
 & & & \downarrow\searrow & \ddots & \searrow\downarrow & & \\
(& 1 & 2 & 4 & 4 & 4 & 4 &) \\
 & & \downarrow\searrow & \searrow\downarrow & & & & \\
(& 1 & 3 & 3 & 3 & 4 & 4 &) \\
 & \downarrow\searrow & \searrow\downarrow & & & & & \\
(& 2 & 2 & 2 & 3 & 4 & 4 &) \\
 & & & \downarrow\searrow & \ddots & \searrow\downarrow & & \\
(& 2 & 2 & 3 & 3 & 3 & 3 &)
\end{array}
\qquad
\begin{array}{cccccccc}
(& 1 & 2 & 3 & 4 & 5 & 5 &) \\
 & & \downarrow\searrow & \ddots & \ddots & \searrow\downarrow & & \\
(& 1 & 3 & 3 & 3 & 4 & 4 &) \\
 & \downarrow\searrow & \ddots & \ddots & \ddots & \searrow\downarrow & & \\
(& 2 & 2 & 3 & 3 & 3 & 3 &)
\end{array}
$$

Figure 3: Derivations of tree path-length sequences via balancing exchanges for $n = 6$

Theorem 5 *The path-length imbalance ordering is a partial order.*

Proof
It is reflexive and transitive by construction. Also the imbalance ordering is antisymmetric, because $\mathbf{s} \trianglelefteq \mathbf{t}$ and $\mathbf{t} \trianglelefteq \mathbf{s}$ together imply $\mathbf{s} = \mathbf{t}$. Otherwise there would be a sequence of balancing exchanges that transform **t** to **s** and ultimately back to **t**; this is not possible, as each balancing exchange reduces by at least 1 the sum of the values in the sequence. □

The imbalance partial order is straightforward to derive for small values of n. In Figure 4, it is displayed for $n = 4, 5, 6, 7$. The most imbalanced sequence appears at the top of the partial order, and an edge from a sequence **s** down to another **t** means that a balancing exchange is possible from **s** to **t**.

It is evident from Figure 4 that the minimal exchanges define the bulk of the ordering. In order to get a deeper appreciation for its structure, Figure 5 presents the ordering for $n = 6, 7, 8, 9$. For clarity Figure 5 presents the ordering in **transitively reduced** format, i.e., as a minimal set of edges omitting those that appear in the transitive closure of the minimal set.

Figures 4 and 5 suggest a number of results about the imbalance ordering. For example they show, and even suggest a proof, that the ordering for any particular value of n is a subordering of that for $n + 1$. They also suggest the following result.

```
1233           12344                  123455                  1234566
 |               |  \                /  |   \                /   |  \\
 |               |   \              /   |    \              /    |   \\
 |               |    \            /    |     \            /     |    \\
2222           13333   )          /   124444   )          /   1235555  )\
                 |    /          /    /|     /          /    /  |   \/  \
                 |   /          (    /  |    /          (    /   |   /\   \
                 |  /            \  /   |   /            \  /    |  /  \   \
               22233              \(  133344              \(  1244455   \   \
                                   \\   |   \              \\/   |  \\   \   \
                                    \\  |    \              \\   |   \\   \   \
                                     \\ |     \            / \\  |    \\   \   \
                                      222344   )          /   1333455  )\   \   \
                                        |     /          /   /   |   \/  \   \   )
                                        |    /          (   /    |   /\   \   ) /
                                        |   /            \ /     |  /  \   \ / /
                                      223333              \   1334444   2223455
                                                         / \  /  |     /   /
                                                        (   \/   |    /   /
                                                         \  /\   |   /   /
                                                          \(  2224444   /
                                                           \\    |     /
                                                            \\   |    /
                                                             \\  |   /
                                                              2233344
                                                                 |
                                                                 |
                                                                 |
                                                              2333333
```

Figure 4: The path-length imbalance ordering for $n = 4,\ 5,\ 6,\ 7$; edges denote ternary exchanges

Theorem 6 *A sequence is on level k of the imbalance partial order (counting down from 0, the topmost and least balanced level)* **iff** *k minimal balancing exchanges are needed to derive it from the least balanced sequence ($1\ 2\ 3\ \ldots\ (n-2)\ (n-1)\ (n-1)$). In this situation the sum of the values in the sequence is*

$$\frac{(n+2)(n-1)}{2} \quad - \quad k.$$

Thus the level of a sequence in the partial order is determined by the sum of its path-length values.

Proof
By induction on k. For the basis $k = 0$, the sum of the path-lengths in the least balanced sequence is $\left(\sum_{i=1}^{n-1} i\right) + (n-1) = (n+2)(n-1)/2$. For the induction step, consider a sequence whose sum of values is $\frac{(n+2)(n-1)}{2} - k$ with $k > 0$. By Theorem 3, this sequence must contain at least three identical values ($q\,q\,q$). Thus there is a minimal balancing exchange to this sequence from another that contains ($(q-1)\,(q+1)\,(q+1)$). This sequence is at level $k-1$ by construction, and by induction it has the stated sum. □

Theorem 6 shows the significance of the level of a sequence in the imbalance partial order:

Definition 2 *The* **level of balance** *of a path-length sequence* **s** *is*

$$\textbf{level of balance of s} \quad = \quad \frac{(n+2)(n-1)}{2} \quad - \quad (\textit{sum of the path-length values in } \mathbf{s}).$$

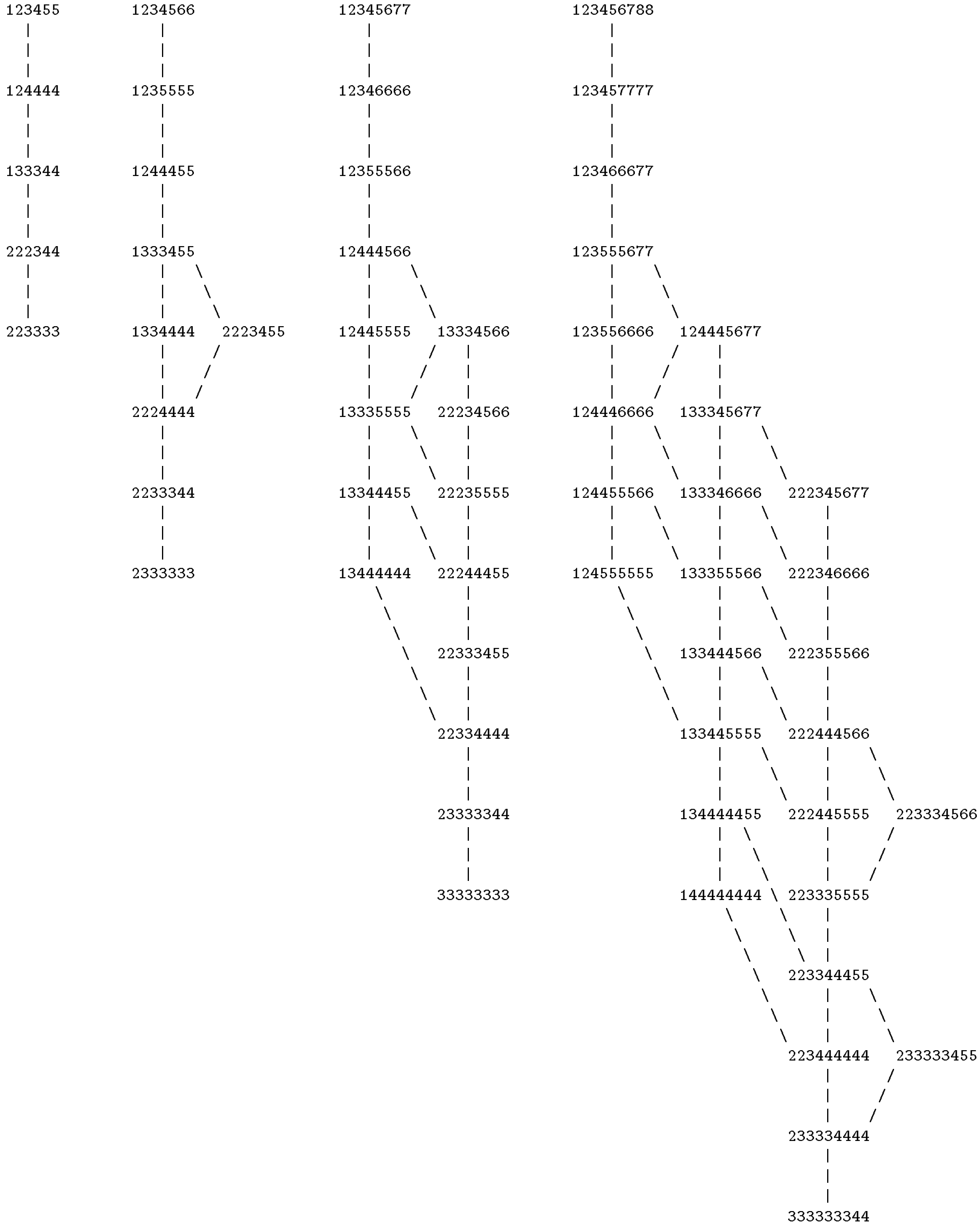

Figure 5: The (transitively reduced) path-length imbalance ordering for $n = 6,\ 7,\ 8,\ 9$

3.3 The Majorization Lattice

We reproduce basic majorization concepts developed in the paper [21]. Majorization as defined here is an extension of the classical majorization of Muirhead and Hardy-Littlewood-Pólya [8], which they used in the study of inequalities, The book by Marshall and Olkin [16] provides a very good account of the classical theory and its applications. The classical theory defines a majorization ordering on descendingly-ordered (or sometimes ascendingly-ordered) multisets, and although quite beautiful is also quite complex. We have transplanted the theory to rely only on linear algebra and convexity. Thus the definitions in this section are ours, and the results vary from those in [16].

Definition 3 *Let $\leq^+$ be the element-wise ordering on sequences so that*

$$\mathbf{x} \;\leq^+\; \mathbf{y} \quad \textbf{iff} \quad x_i \;\leq\; y_i \;\; \textit{for all } i.$$

Definition 4 *The* **Zeta matrix** *$S = (\zeta_{ij})$ is defined by*

$$\zeta_{ij} \;=\; \textit{1 if } i \geq j, \quad \textit{0 otherwise.}$$

Theorem 7 *If $\mathbf{x}$ and $\mathbf{y}$ are density sequences (so $\sum_{i=1}^n x_i = \sum_{i=1}^n y_i = 1$), then*

$$\mathbf{x} \;\preceq\; \mathbf{y} \quad \textbf{iff} \quad (S\,\mathbf{x}) \;\leq^+\; (S\,\mathbf{y}).$$

Proof: When $\mathbf{x}$ and $\mathbf{y}$ are density sequences, $(S\,\mathbf{x}) \leq^+ (S\,\mathbf{y})$ is equivalent to the following:

$$\begin{array}{rcl} x_1 & \leq & y_1 \\ x_1 + x_2 & \leq & y_1 + y_2 \\ & \vdots & \\ x_1 + x_2 + \cdots + x_n & \leq & y_1 + y_2 + \cdots + y_n. \end{array}$$

Thus $\mathbf{x}$ is a density sequence **iff** $(S\,\mathbf{x})$ is a distribution sequence. □

Definition 5 *The* **Möbius matrix** *$D = (\mu_{ij})$ is defined by*

$$\mu_{ij} \;=\; \textit{1 if } i = j, \quad -1 \textit{ if } j = i-1, \quad \textit{0 otherwise.}$$

The Möbius matrix is the inverse of the Zeta matrix. For example, when $n = 9$:

$$S \;=\; \begin{pmatrix} 1&0&0&0&0&0&0&0&0\\ 1&1&0&0&0&0&0&0&0\\ 1&1&1&0&0&0&0&0&0\\ 1&1&1&1&0&0&0&0&0\\ 1&1&1&1&1&0&0&0&0\\ 1&1&1&1&1&1&0&0&0\\ 1&1&1&1&1&1&1&0&0\\ 1&1&1&1&1&1&1&1&0\\ 1&1&1&1&1&1&1&1&1 \end{pmatrix} \qquad D \;=\; S^{-1} \;=\; \begin{pmatrix} 1&0&0&0&0&0&0&0&0\\ -1&1&0&0&0&0&0&0&0\\ 0&-1&1&0&0&0&0&0&0\\ 0&0&-1&1&0&0&0&0&0\\ 0&0&0&-1&1&0&0&0&0\\ 0&0&0&0&-1&1&0&0&0\\ 0&0&0&0&0&-1&1&0&0\\ 0&0&0&0&0&0&-1&1&0\\ 0&0&0&0&0&0&0&-1&1 \end{pmatrix}.$$

The Möbius matrix is also significant in that it corresponds directly to the concept of pairwise exchange (of adjacent elements in a sequence). The theory of Möbius inversion [22] gives a generalized notion of differential on partially-ordered domains (although here we consider only totally-ordered sequences). Basically we can think of S as an integral operator (which replaces a sequence by its left-to-right "integral"), with D as its inverse differential operator.

Theorem 8 *If $\mathbf{u}$ and $\mathbf{v}$ are distribution sequences, then $D\,\mathbf{u}$ and $D\,\mathbf{v}$ are density sequences and*

$$\mathbf{u} \;\leq^+\; \mathbf{v} \quad \textbf{iff} \quad (D\,\mathbf{u}) \;\preceq\; (D\,\mathbf{v}).$$

Proof: Follows immediately from the previous theorem, as the Moebius and Zeta transformations are inverses of one another. □

This isomorphism between $\leq^+$ and $\preceq$ gives an immediate demonstration that majorization defines a lattice.

Definition 6 *The majorization lub and glb operations are definable as*

$$\begin{aligned} \mathbf{x} \ \sqcup \ \mathbf{y} &= D\,((S\,\mathbf{x}) \ \ \mathbf{max}^+ \ \ (S\,\mathbf{y})) \\ \mathbf{x} \ \sqcap \ \mathbf{y} &= D\,((S\,\mathbf{x}) \ \ \mathbf{min}^+ \ \ (S\,\mathbf{y})). \end{aligned}$$

where $\mathbf{min}^+$ *is vector element-wise minimum, and* $\mathbf{max}^+$ *is vector element-wise maximum.*

Theorem 9 *If $\mathcal{D}$ is the set of density sequences of length n (nonnegative $\mathbf{x}$ with $\sum_{i=1}^{n} x_i = 1$), then $\langle \mathcal{D}, \preceq, \sqcap, \sqcup, \perp, \top \rangle$ forms a complete distributive lattice with least element $\perp = (\,0\,0 \cdots 0\,1\,)$ and greatest element $\top = (\,1\,0 \cdots 0\,0\,)$.*

Proof: Let $\mathcal{P}$ be the set of distribution sequences of length n (ascending nonnegative vectors $\mathbf{v}$ with $v_n = 1$). The transformation $\mathbf{x} \mapsto S\,\mathbf{x}$ defines a lattice isomorphism between $\mathcal{D}$ and $\mathcal{P}$. Specifically, it gives a bijection between the majorization algebra of densities $\langle \mathcal{D}, \preceq, \sqcap, \sqcup, \perp, \top \rangle$ and the complete distributive lattice of distributions $\langle \mathcal{P}, \leq^+, \mathbf{min}^+, \mathbf{max}^+, (\,0\,0 \cdots 0\,1\,), (\,1\,1 \cdots 1\,1\,) \rangle$.

Note that $\mathbf{x} \sqcap \mathbf{y}$ and $\mathbf{x} \sqcup \mathbf{y}$ are defined just so as to be the majorization glb and lub operations:

$\mathbf{z} \preceq \mathbf{x}, \quad \mathbf{z} \preceq \mathbf{y}$	$\mathbf{x} \preceq \mathbf{z}, \quad \mathbf{y} \preceq \mathbf{z}$
iff $(S\,\mathbf{z}) \leq^+ (S\,\mathbf{x}), \quad (S\,\mathbf{z}) \leq^+ (S\,\mathbf{y})$	**iff** $(S\,\mathbf{x}) \leq^+ (S\,\mathbf{z}), \quad (S\,\mathbf{y}) \leq^+ (S\,\mathbf{z})$
iff $(S\,\mathbf{z}) \leq^+ (\,(S\,\mathbf{x})\ \mathbf{min}^+\ (S\,\mathbf{y})\,)$	**iff** $(\,(S\,\mathbf{x})\ \mathbf{max}^+\ (S\,\mathbf{y})\,) \leq^+ (S\,\mathbf{z})$
iff $\mathbf{z} \preceq D\,(\,(S\,\mathbf{x})\ \mathbf{min}^+\ (S\,\mathbf{y})\,)$	**iff** $D\,(\,(S\,\mathbf{x})\ \mathbf{max}^+\ (S\,\mathbf{y})\,) \preceq \mathbf{z}$
iff $\mathbf{z} \preceq \mathbf{x} \sqcap \mathbf{y}$.	**iff** $\mathbf{x} \sqcup \mathbf{y} \preceq \mathbf{z}$.

Thus the majorization algebra of densities is also a complete distributive lattice. □

Even when $\mathbf{x}$ and $\mathbf{y}$ are in descending order, the sequences $(\mathbf{x} \sqcap \mathbf{y})$ and $(\mathbf{x} \sqcap \mathbf{y})$ defined here are not necessarily in descending order. For example, $\mathbf{x} = (\,2^{-2}\,2^{-2}\,2^{-3}\,2^{-4}\,2^{-4}\,2^{-4}\,2^{-4}\,2^{-4}\,2^{-4}\,)$ and $\mathbf{y} = (\,2^{-2}\,2^{-3}\,2^{-3}\,2^{-3}\,2^{-3}\,2^{-3}\,2^{-4}\,2^{-5}\,2^{-5}\,)$ together give the least upper bound value $\mathbf{x} \sqcup \mathbf{y} = D\,((S\,\mathbf{x})\,\mathbf{max}^+\,(S\,\mathbf{y})) = (\,2^{-2}\,2^{-2}\,2^{-3}\,\boxed{2^{-4}}\,2^{-4}\,\boxed{2^{-3}}\,2^{-4}\,2^{-5}\,2^{-5}\,)$. See Figure 8.

3.4 The Imbalance Lattice: A Discrete Cousin of the Majorization Lattice

Since every pair of sequences in Figures 4 and 5 has a unique glb and lub (meet and join), it appears that the imbalance ordering is not only a partial order, but also a *lattice*.

In this section we prove that this impression is correct. and relate the **imbalance lattice** directly to the majorization lattice. This approach clarifies the role of majorization in Huffman coding, and provides a way to evaluate functions on the imbalance lattice.

As mentioned earlier, every pair of sequences $\mathbf{s}$, $\mathbf{t}$ in Figures 4 and 5 has a unique glb and lub (meet and join), which we can write $\mathbf{s} \wedge \mathbf{t}$, $\mathbf{s} \vee \mathbf{t}$. For example when $n = 9$, $\mathbf{s} = (\,1\,2\,4\,4\,5\,5\,5\,6\,6\,)$ and $\mathbf{t} = (\,2\,2\,2\,3\,4\,5\,6\,7\,7\,)$ have $\mathbf{s} \wedge \mathbf{t} = (\,2\,2\,2\,3\,5\,5\,5\,6\,6\,)$, $\mathbf{s} \vee \mathbf{t} = (\,1\,2\,4\,4\,4\,5\,6\,7\,7\,)$. See Figure 6.

An example showing how closely related the Huffman and majorization lattices are is in Figure 7. It studies the tree path-length sequences $\mathbf{s}$, $\mathbf{t}$ just mentioned. For this pair, the glb and lub (meet and join) $2^{-\mathbf{s}} \sqcap 2^{-\mathbf{t}}$, $2^{-\mathbf{s}} \sqcup 2^{-\mathbf{t}}$ in the majorization lattice correspond precisely to $\mathbf{s} \wedge \mathbf{t}$, $\mathbf{s} \vee \mathbf{t}$ in the imbalance lattice as

$$2^{-\mathbf{s}} \sqcap 2^{-\mathbf{t}} \;=\; 2^{-(\mathbf{s}\,\wedge\,\mathbf{t})}, \qquad 2^{-\mathbf{s}} \sqcup 2^{-\mathbf{t}} \;=\; 2^{-(\mathbf{s}\,\vee\,\mathbf{t})} \qquad \text{(but beware: this does not always hold).}$$

Theorem 10 is the key result relating imbalance and majorization.

```
            123556666   124445677   =   (s \/ t)
                  \       /   \
                   \     /     \
                    \   /       \
                  124446666   133345677
                    /   \       /   \
                   /     \     /     \
                  /       \   /       \
    s   =   124455566   133346666   222345677   =   t
              /   \       /   \       /
             /     \     /     \     /
            /       \   /       \   /
      124555555   133355566   222346666
          |         /   \       /
          |        /     \     /
          |       /       \   /
          | 133444566   222355566   =   (s /\ t)
          |   /   \       /
          |  /     \     /
          | /       \   /
      133445555   222444566
        /   \       /   \
       /     \     /     \
      /       \   /       \
134444455   222445555   223334566
```

Figure 6: Part of the imbalance lattice for $n = 9$ redrawn to highlight lower and upper bounds

$$
\begin{array}{rcrl}
\mathbf{s} & = & & (\ 1 \quad 2 \quad 4 \quad 4 \quad 5 \quad 5 \quad 5 \quad 6 \quad 6\ ) \\
2^{-\mathbf{s}} & = & & (\ 2^{-1} \quad 2^{-2} \quad 2^{-4} \quad 2^{-4} \quad 2^{-5} \quad 2^{-5} \quad 2^{-5} \quad 2^{-6} \quad 2^{-6}\ ) \\
S\,2^{-\mathbf{s}} & = & 2^{-7} & (\ 64 \quad 96 \quad 104 \quad 112 \quad 116 \quad 120 \quad 124 \quad 126 \quad 128\ ) \\
\\
\mathbf{t} & = & & (\ 2 \quad 2 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 7\ ) \\
2^{-\mathbf{t}} & = & & (\ 2^{-2} \quad 2^{-2} \quad 2^{-2} \quad 2^{-3} \quad 2^{-4} \quad 2^{-5} \quad 2^{-6} \quad 2^{-7} \quad 2^{-7}\ ) \\
S\,2^{-\mathbf{t}} & = & 2^{-7} & (\ 32 \quad 64 \quad 96 \quad 112 \quad 120 \quad 124 \quad 126 \quad 127 \quad 128\ ) \\
\\
(S\,2^{-\mathbf{s}})\ \mathbf{max}^{+}\ (S\,2^{-\mathbf{t}}) & = & 2^{-7} & (\ 64 \quad 96 \quad 104 \quad 112 \quad 120 \quad 124 \quad 126 \quad 127 \quad 128\ ) \\
2^{-\mathbf{s}} \sqcup 2^{-\mathbf{t}} \ = \ D\,((S\,2^{-\mathbf{s}})\ \mathbf{max}^{+}\ (S\,2^{-\mathbf{t}})) & = & 2^{-7} & (\ 64 \quad 32 \quad 8 \quad 8 \quad 8 \quad 4 \quad 2 \quad 1 \quad 1\ ) \\
 & = & & (\ 2^{-1} \quad 2^{-2} \quad 2^{-4} \quad 2^{-4} \quad 2^{-4} \quad 2^{-5} \quad 2^{-6} \quad 2^{-7} \quad 2^{-7}\ ) \\
\mathbf{s} \vee \mathbf{t} & = & & (\ 1 \quad 2 \quad 4 \quad 4 \quad 4 \quad 5 \quad 6 \quad 7 \quad 7\ ) \\
\\
(S\,2^{-\mathbf{s}})\ \mathbf{min}^{+}\ (S\,2^{-\mathbf{t}}) & = & 2^{-7} & (\ 32 \quad 64 \quad 96 \quad 112 \quad 116 \quad 120 \quad 124 \quad 126 \quad 128\ ) \\
2^{-\mathbf{s}} \sqcap 2^{-\mathbf{t}} \ = \ D\,((S\,2^{-\mathbf{s}})\ \mathbf{min}^{+}\ (S\,2^{-\mathbf{t}})) & = & 2^{-7} & (\ 32 \quad 32 \quad 32 \quad 16 \quad 4 \quad 4 \quad 4 \quad 2 \quad 2\ ) \\
 & = & & (\ 2^{-2} \quad 2^{-2} \quad 2^{-2} \quad 2^{-3} \quad 2^{-5} \quad 2^{-5} \quad 2^{-5} \quad 2^{-6} \quad 2^{-6}\ ) \\
\mathbf{s} \wedge \mathbf{t} & = & & (\ 2 \quad 2 \quad 2 \quad 3 \quad 5 \quad 5 \quad 5 \quad 6 \quad 6\ )
\end{array}
$$

Figure 7: Related points in the majorization and imbalance lattices, showing their close connection

$$
\begin{array}{rcrcl}
 & & \mathbf{s} & = & (\; 2 \quad 2 \quad 3 \quad 4 \quad 4 \quad 4 \quad 4 \quad 4 \quad 4 \;) \\
 & & 2^{-\mathbf{s}} & = & (\; 2^{-2} \quad 2^{-2} \quad 2^{-3} \quad 2^{-4} \quad 2^{-4} \quad 2^{-4} \quad 2^{-4} \quad 2^{-4} \quad 2^{-4} \;) \\
 & & S\,2^{-\mathbf{s}} & = & 2^{-7}\,(\; 32 \quad 64 \quad 80 \quad 88 \quad 96 \quad 104 \quad 112 \quad 120 \quad 128 \;) \\
 & & & & \\
 & & \mathbf{t} & = & (\; 2 \quad 3 \quad 3 \quad 3 \quad 3 \quad 3 \quad 4 \quad 5 \quad 5 \;) \\
 & & 2^{-\mathbf{t}} & = & (\; 2^{-2} \quad 2^{-3} \quad 2^{-3} \quad 2^{-3} \quad 2^{-3} \quad 2^{-3} \quad 2^{-4} \quad 2^{-5} \quad 2^{-5} \;) \\
 & & S\,2^{-\mathbf{t}} & = & 2^{-7}\,(\; 32 \quad 48 \quad 64 \quad 80 \quad 96 \quad 112 \quad 120 \quad 124 \quad 128 \;) \\
 & & & & \\
 & & (S\,2^{-\mathbf{s}})\ \mathbf{max}^{+}\ (S\,2^{-\mathbf{t}}) & = & 2^{-7}\,(\; 32 \quad 64 \quad 80 \quad 88 \quad 96 \quad 112 \quad 120 \quad 124 \quad 128 \;) \\
2^{-\mathbf{s}} \sqcup 2^{-\mathbf{t}} & = & D\,((S\,2^{-\mathbf{s}})\ \mathbf{max}^{+}\ (S\,2^{-\mathbf{t}})) & = & 2^{-7}\,(\; 32 \quad 32 \quad 16 \quad 8 \quad 8 \quad 16 \quad 8 \quad 4 \quad 4 \;) \\
 & & & = & (\; 2^{-2} \quad 2^{-2} \quad 2^{-3} \quad \boxed{2^{-4}} \quad 2^{-4} \quad \boxed{2^{-3}} \quad 2^{-4} \quad 2^{-5} \quad 2^{-5} \;) \\
 & & \mathbf{s} \vee \mathbf{t} & = & (\; 2 \quad 2 \quad 3 \quad \boxed{3} \quad 4 \quad \boxed{4} \quad 4 \quad 5 \quad 5 \;)
\end{array}
$$

Figure 8: Results of $(2^{-\mathbf{s}} \sqcap 2^{-\mathbf{t}})$ and $(2^{-\mathbf{s}} \sqcup 2^{-\mathbf{t}})$ are not necessarily in ascending order

$$
\begin{array}{rcrcl}
 & & \mathbf{s} & = & (\; 1 \quad 2 \quad 4 \quad 5 \quad 5 \quad 5 \quad 5 \quad 5 \quad 5 \;) \\
 & & 2^{-\mathbf{s}} & = & (\; 2^{-1} \quad 2^{-2} \quad 2^{-4} \quad 2^{-5} \quad 2^{-5} \quad 2^{-5} \quad 2^{-5} \quad 2^{-5} \quad 2^{-5} \;) \\
 & & S\,2^{-\mathbf{s}} & = & 2^{-7}\,(\; 64 \quad 96 \quad 104 \quad 108 \quad 112 \quad 116 \quad 120 \quad 124 \quad 128 \;) \\
 & & & & \\
 & & \mathbf{t} & = & (\; 2 \quad 2 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 7 \;) \\
 & & 2^{-\mathbf{t}} & = & (\; 2^{-2} \quad 2^{-2} \quad 2^{-2} \quad 2^{-3} \quad 2^{-4} \quad 2^{-5} \quad 2^{-6} \quad 2^{-7} \quad 2^{-7} \;) \\
 & & S\,2^{-\mathbf{t}} & = & 2^{-7}\,(\; 32 \quad 64 \quad 96 \quad 112 \quad 120 \quad 124 \quad 126 \quad 127 \quad 128 \;) \\
 & & & & \\
 & & (S\,2^{-\mathbf{s}})\ \mathbf{min}^{+}\ (S\,2^{-\mathbf{t}}) & = & 2^{-7}\,(\; 32 \quad 64 \quad 96 \quad 108 \quad 112 \quad 116 \quad 120 \quad 124 \quad 128 \;) \\
2^{-\mathbf{s}} \sqcap 2^{-\mathbf{t}} & = & D\,((S\,2^{-\mathbf{s}})\ \mathbf{min}^{+}\ (S\,2^{-\mathbf{t}})) & = & 2^{-7}\,(\; 32 \quad 32 \quad 32 \quad 12 \quad 4 \quad 4 \quad 4 \quad 4 \quad 4 \;) \\
 & & & = & (\; 2^{-2} \quad 2^{-2} \quad 2^{-2} \quad \boxed{2^{-\alpha}} \quad \boxed{2^{-5}} \quad 2^{-5} \quad 2^{-5} \quad 2^{-5} \quad 2^{-5} \;) \\
 & & & & \qquad\qquad -\alpha \;=\; (-7 + \log_2(12)) \;\approx\; -3.4150375 \\
 & & & \succeq & (\; 2^{-2} \quad 2^{-2} \quad 2^{-2} \quad \boxed{2^{-4}} \quad \boxed{2^{-4}} \quad 2^{-5} \quad 2^{-5} \quad 2^{-5} \quad 2^{-5} \;) \\
 & & \mathbf{s} \wedge \mathbf{t} & = & (\; 2 \quad 2 \quad 2 \quad \boxed{4} \quad \boxed{4} \quad 5 \quad 5 \quad 5 \quad 5 \;)
\end{array}
$$

Here $2^{-(\mathbf{s} \wedge \mathbf{t})} \preceq (2^{-\mathbf{s}} \sqcap 2^{-\mathbf{t}})$, and the two differ in the indicated positions. See the lattice in Figure 6.

Figure 9: The imbalance lattice is not simply conjugate to a sublattice of the majorization lattice

Theorem 10 *On tree path-length sequences, the imbalance ordering is isomorphic to the majorization ordering. Specifically, whenever* $\mathbf{s}$ *and* $\mathbf{t}$ *are tree path-length sequences, then*

$$\mathbf{s} \;\trianglelefteq\; \mathbf{t} \quad \textbf{iff} \quad 2^{-\mathbf{s}} \;\preceq\; 2^{-\mathbf{t}}.$$

Proof
We show first that every balancing exchange produces a reduction in the majorization ordering. Let $\mathbf{s}$ be the result of balancing exchange on $\mathbf{t}$ (so $\mathbf{s} \trianglelefteq \mathbf{t}$). Then:

$$\begin{array}{rcllllllllllll}
\mathbf{t} & = & (\,\cdot & \cdots & \cdot & p & u & \cdot & \cdots & v & (q+1) & (q+1) & \cdots & \cdot\,) \\
\mathbf{s} & = & (\,\cdot & \cdots & \cdot & (p+1) & (p+1) & u & \cdots & \cdot & v & q & \cdots & \cdot\,) \\
 & & & & & & & & & & & & & \\
2^{-\mathbf{t}} & = & (\,\cdot & \cdots & \cdot & 2^{-p} & 2^{-u} & \cdot & \cdots & 2^{-v} & 2^{-(q+1)} & 2^{-(q+1)} & \cdots & \cdot\,) \\
2^{-\mathbf{s}} & = & (\,\cdot & \cdots & \cdot & 2^{-(p+1)} & 2^{-(p+1)} & 2^{-u} & \cdots & \cdot & 2^{-v} & 2^{-q} & \cdots & \cdot\,) \\
 & & & & & & & & & & & & & \\
S\,2^{-\mathbf{t}} & = & (\,\cdot & \cdots & S & S{+}2^{-p} & S{+}2^{-p}{+}2^{-u} & \cdot & \cdots & T & T{+}2^{-(q+1)} & T{+}2^{-q} & \cdots & 1\,) \\
S\,2^{-\mathbf{s}} & = & (\,\cdot & \cdots & S & S{+}2^{-(p+1)} & S{+}2^{-p} & \cdot & \cdots & T{-}2^{-v} & T & T{+}2^{-q} & \cdots & 1\,) \\
\hline
S\,2^{-\mathbf{t}} - S\,2^{-\mathbf{s}} & = & (\,0 & \cdots & 0 & +2^{-(p+1)} & +2^{-u} & \cdot & \cdots & +2^{-v} & +2^{-(q+1)} & 0 & \cdots & 0\,)
\end{array}$$

Thus $S\,2^{-\mathbf{s}}$ and $S\,2^{-\mathbf{t}}$ differ only in the values appearing between p and q, and each element in $S\,2^{-\mathbf{t}} - S\,2^{-\mathbf{s}}$ is nonnegative, so $2^{-\mathbf{s}} \preceq 2^{-\mathbf{t}}$.

The proof of the converse, that $2^{-\mathbf{s}} \preceq 2^{-\mathbf{t}}$ implies $\mathbf{s} \trianglelefteq \mathbf{t}$ for tree path-length sequences $\mathbf{s}$, $\mathbf{t}$, can proceed by induction on m, the difference in the levels of balance of $\mathbf{s}$ and $\mathbf{t}$:

$$m \quad = \quad (\text{level of balance of } \mathbf{t}) \;-\; (\text{level of balance of } \mathbf{s}).$$

By Theorem 4, two unequal sequences differ in at least three values, and if they differ in three values they differ by a balancing exchange. The basis $m = 1$ is then the balancing exchange case discussed just above. By induction, if $m > 1$ and $2^{-\mathbf{s}} \preceq 2^{-\mathbf{t}}$ let a, b, c, d be the extreme (outermost) pairwise-differing values among the two sorted sequences, so that $\mathbf{s} = (\cdots\ a\ \cdots\ b\ \cdots)$ and $\mathbf{t} = (\cdots\ c\ \cdots\ d\ \cdots)$, and $c < a$, $b < d$ because of the majorization inequality, and $a \le b$ and $c \le d$ because the sequences are sorted. Thus $c \ne d$ since $c < a \le b < d$. By the Kraft equality $2^{-a} + \cdots + 2^{-b} \;=\; 2^{-c} + \cdots + 2^{-d}$, and because $b < d$ necessarily $\mathbf{t} = (\cdots\ c\ \cdots\ d\ d\ \cdots)$, since otherwise we reach a contradiction (multiplying both sides of the equality by 2^d makes the left side even but the right side odd). Thus, if we define $\mathbf{t}' = (\cdots\ (c+1)\ (c+1)\ \cdots\ (d-1)\ \cdots)$, the result of a balancing exchange on $\mathbf{t} = (\cdots\ c\ \cdots\ d\ d\ \cdots)$, then the level difference between $\mathbf{s}$ and $\mathbf{t}'$ is at most $(m-1)$, and $2^{-\mathbf{t}'} \preceq 2^{-\mathbf{t}}$. Furthermore we claim $2^{-\mathbf{s}} \preceq 2^{-\mathbf{t}'}$, using the following analysis:

$$\begin{array}{rclllllllll}
\mathbf{t} & = & (\,\cdot & \cdots & \cdot & c & u & \cdots & d & d & \cdots\ \cdot\,) \\
\mathbf{t}' & = & (\,\cdot & \cdots & \cdot & (c+1) & (c+1) & \cdots & \cdot & (d-1) & \cdots\ \cdot\,) \\
\mathbf{s} & = & (\,\cdot & \cdots & \cdot & a & w & \cdots & \cdot & b & \cdots\ \cdot\,) \\
 & & & & & & & & & & \\
2^{-\mathbf{t}} & = & (\,\cdot & \cdots & \cdot & 2^{-c} & 2^{-u} & \cdots & 2^{-d} & 2^{-d} & \cdots\ \cdot\,) \\
2^{-\mathbf{t}'} & = & (\,\cdot & \cdots & \cdot & 2^{-(c+1)} & 2^{-(c+1)} & \cdots & \cdot & 2^{-(d-1)} & \cdots\ \cdot\,) \\
2^{-\mathbf{s}} & = & (\,\cdot & \cdots & \cdot & 2^{-a} & 2^{-w} & \cdots & \cdot & 2^{-b} & \cdots\ \cdot\,) \\
 & & & & & & & & & & \\
S\,2^{-\mathbf{t}} - S\,2^{-\mathbf{s}} & = & (\,0 & \cdots & 0 & +(2^{-c} - 2^{-a}) & +S_1 & \cdots & +S_k & 0 & \cdots\ 0\,) \\
S\,2^{-\mathbf{t}} - S\,2^{-\mathbf{t}'} & = & (\,0 & \cdots & 0 & +2^{-(c+1)} & +2^{-u} & \cdots & +2^{-d} & 0 & \cdots\ 0\,) \\
\hline
S\,2^{-\mathbf{t}'} - S\,2^{-\mathbf{s}} & = & (\,0 & \cdots & 0 & +(2^{-(c+1)} - 2^{-a}) & +(S_1 - 2^{-u}) & \cdots & +(S_k - 2^{-d}) & 0 & \cdots\ 0\,)
\end{array}$$

Because $2^{-\mathbf{s}} \preceq 2^{-\mathbf{t}}$, the running totals $S_1, \ldots, S_k$ are nonnegative. Also, $(2^{-(c+1)} - 2^{-a}) \ge 0$ since $c < a$. Furthermore $c \le (a-1) \le w$, implying $S_1 - 2^{-u} \;=\; (2^{-c} - 2^{-a}) - 2^{w} \ge 2^{-(a-1)} - 2^{-w} \ge 0$. Finally $S_k + 2^{-d} - 2^{-b} = 0$, so $b \le (d-1)$ implies $S_k - 2^{-d} \;=\; (2^{-b} - 2^{-d}) - 2^{-d} \;\ge\; 2^{-b} - 2^{-(d-1)} \;\ge\; 0$. Thus $S\,2^{-\mathbf{s}} \le^{+} S\,2^{-\mathbf{t}'}$ (i.e., $2^{-\mathbf{s}} \preceq 2^{-\mathbf{t}'}$), and induction completes the proof. $\square$

Definition 7 *Let* $\boldsymbol{\ell} = (\,\ell_1 \cdots \ell_n\,)$ *be a tree path-length sequence of length* n.

The **contraction** $\widehat{\boldsymbol{\ell}}$ *of* $\boldsymbol{\ell}$ *is the sequence of length* $(n-1)$ *defined by*

$$\widehat{\boldsymbol{\ell}} \quad = \quad \mathrm{sort}\!\uparrow[(\;\ell_1\;\cdots\;\ell_{n-2}\;\;(\ell_{n-1}-1)\;)].$$

The **position** i **expansion of** $\boldsymbol{\ell}$ *is the sequence of length* $(n+1)$ *defined by*

$$\mathrm{sort}\!\uparrow[(\;\ell_1\;\cdots\;\ell_{i-1}\;(\ell_i+1)\;(\ell_i+1)\;\ell_{i+1}\;\cdots\;\ell_n\;)].$$

If we write

$$\boldsymbol{\ell} \quad = \quad (\;\cdots\;(q-j)\;\overbrace{q\;\cdots\;q}^{2k}\;)$$

with $j,\,k > 0$ *(by Theorem 2), then we call* $2k$ *the* **suffix length** *of* $\boldsymbol{\ell}$*, and* j *its* **suffix increment***.* *The* **contraction** *of* $\boldsymbol{\ell}$ *is then*

$$\widehat{\boldsymbol{\ell}} \quad = \quad (\;\cdots\;(q-j)\;\;(q-1)\;\;\overbrace{q\;\cdots\;q}^{2k-2}\;).$$

The **upper expansion** $\boldsymbol{\ell}^+$ *is the position* n *expansion of* $\boldsymbol{\ell}$*:*

$$\boldsymbol{\ell}^+ \quad = \quad (\;\cdots\;(q-j)\;\;\overbrace{q\;\cdots\;q}^{2k-1}\;(q+1)\;\;(q+1)\;).$$

The **lower expansion** $\boldsymbol{\ell}_+$ *is the position* $n-2k$ *expansion of* $\boldsymbol{\ell}$*:*

$$\boldsymbol{\ell}_+ \quad = \quad (\;\cdots\;(q-j+1)\;\;(q-j+1)\;\;\overbrace{q\;\cdots\;q}^{2k}\;).$$

Theorem 11 *If* $\boldsymbol{\ell}$ *is a path-length sequence, then* $\boldsymbol{\ell}_+ \;\trianglelefteq\; \boldsymbol{\ell}^+$. *Furthermore, either* $\boldsymbol{\ell} = (\widehat{\boldsymbol{\ell}})_+$*, or* $\boldsymbol{\ell} = (\widehat{\boldsymbol{\ell}})^+$*. Thus* $(\widehat{\boldsymbol{\ell}})_+ \;\trianglelefteq\; \boldsymbol{\ell} \;\trianglelefteq\; (\widehat{\boldsymbol{\ell}})^+$.

Proof

$\boldsymbol{\ell}_+$ and $\boldsymbol{\ell}^+$ differ by a ternary exchange, so $\boldsymbol{\ell}_+ \trianglelefteq \boldsymbol{\ell}^+$. Theorem 2 proves $\boldsymbol{\ell} = (\cdots(q-j)\,\overbrace{q\cdots q}^{2k})$, and $(\widehat{\boldsymbol{\ell}})^+ = \boldsymbol{\ell}$ if $k=1$ (necessitating $j=1$) and $(\widehat{\boldsymbol{\ell}})_+ = \boldsymbol{\ell}$ if $k>1$, so $\boldsymbol{\ell} \in \{\,(\widehat{\boldsymbol{\ell}})_+,\,(\widehat{\boldsymbol{\ell}})^+\,\}$. □

Theorem 12 *If* $\mathbf{s} \trianglelefteq \mathbf{t}$*, then:* $\widehat{\mathbf{s}} \trianglelefteq \widehat{\mathbf{t}}$*,* $\mathbf{s}_+ \trianglelefteq \mathbf{t}_+$*, and* $\mathbf{s}^+ \trianglelefteq \mathbf{t}^+$.

Proof

Recall that if $\mathbf{s} \trianglelefteq \mathbf{t}$ then there are sequences $\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_m$ $(m \geq 1)$ where $\mathbf{t} = \boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_m = \mathbf{s}$ and for each i, $1 \leq i < m$, there is a balancing exchange from $\boldsymbol{\ell}_i$ to $\boldsymbol{\ell}_{i+1}$. Thus, for example, we can prove $\mathbf{s}^+ \trianglelefteq \mathbf{t}^+$ by converting the derivation $\mathbf{t} = \boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_m = \mathbf{s}$ directly to the derivation $\mathbf{t}^+ = {\boldsymbol{\ell}_1}^+,\ldots,{\boldsymbol{\ell}_m}^+ = \mathbf{s}^+$.
To prove $\widehat{\mathbf{s}} \trianglelefteq \widehat{\mathbf{t}}$ it is enough to show that either the step from $\widehat{\boldsymbol{\ell}_i}$ to $\widehat{\boldsymbol{\ell}_{i+1}}$ is a balancing exchange, or $\widehat{\boldsymbol{\ell}_i} = \widehat{\boldsymbol{\ell}_{i+1}}$. The former must hold if $\boldsymbol{\ell}_i$ and $\boldsymbol{\ell}_{i+1}$ agree in the final two positions. If they disagree,

$$\boldsymbol{\ell}_i \quad = \quad (\;\cdots\;p\;a\;\cdots\;b\;q\;q\;), \qquad \boldsymbol{\ell}_{i+1} \quad = \quad (\;\cdots\;(p+1)\;(p+1)\;a\;\cdots\;b\;(q-1)\;)$$

because they define a balancing exchange, and by Theorem 2 necessarily $b = (q-1)$. But then

$$\widehat{\boldsymbol{\ell}_i} \;=\; \mathrm{sort}\!\uparrow[(\;\cdots\;p\;a\;\cdots\;(q-1)\;(q-1)\;)], \qquad \widehat{\boldsymbol{\ell}_{i+1}} \;=\; \mathrm{sort}\!\uparrow[(\;\cdots\;(p+1)\;(p+1)\;a\;\cdots\;(q-2)\;)].$$

If $(p+1) = b = (q-1)$, then $p = (q-2)$ and the two contractions are equal. If not, they still differ by a balancing exchange.
The proof that $\mathbf{s}_+ \trianglelefteq \mathbf{t}_+$ is extremely similar, where $(\boldsymbol{\ell}_i)_+ = (\boldsymbol{\ell}_{i+1})_+$ iff $\boldsymbol{\ell}_i = (\;\cdots\;(q-j)\;q\;q\;\cdots\;q\;)$ $\boldsymbol{\ell}_{i+1} = (\;\cdots\;(q-j+1)\;(q-j+1)\;(q-1)\;\cdots\;q\;)$ where $j \geq 2$. The proof of $\mathbf{s}^+ \trianglelefteq \mathbf{t}^+$ is also extremely similar, but even easier, since then it is never the case that $(\boldsymbol{\ell}_i)^+ = (\boldsymbol{\ell}_{i+1})^+$. □

Theorem 13 *The imbalance ordering on binary trees determines a bona-fide lattice. That is, whenever* $\mathbf{s}$ *and* $\mathbf{t}$ *are tree path-length sequences of length* n*, then there are unique path-length sequences* $\mathbf{s} \wedge \mathbf{t}$ *and* $\mathbf{s} \vee \mathbf{t}$ *such that:*

- $\mathbf{s} \wedge \mathbf{t} \trianglelefteq \mathbf{s},\ \mathbf{t}$ *and, if* $\boldsymbol{\ell}$ *is any path-length sequence, then* $\boldsymbol{\ell} \trianglelefteq \mathbf{s},\ \mathbf{t}$ **iff** $\boldsymbol{\ell} \trianglelefteq \mathbf{s} \wedge \mathbf{t}$.
- $\mathbf{s},\ \mathbf{t} \trianglelefteq \mathbf{s} \vee \mathbf{t}$ *and, if* $\boldsymbol{\ell}$ *is any path-length sequence, then* $\mathbf{s},\ \mathbf{t} \trianglelefteq \boldsymbol{\ell}$ **iff** $\mathbf{s} \wedge \mathbf{t} \trianglelefteq \boldsymbol{\ell}$.

Proof

By induction on n. We consider only the glb here, the proof for the lub being similar. The theorem holds trivially for $n \leq 6$, since then the trees are totally ordered. Assume it holds for sequences of size $n-1$ or less.

First, $\mathbf{s}$ and $\mathbf{t}$ must have a common lower bound:
The glb $\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}}$ exists by induction, and the lower expansion then gives a lower bound

$$(\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}})_+ \quad \trianglelefteq \quad \mathbf{s},\ \mathbf{t}.$$

Second, if $\mathbf{s}$ and $\mathbf{t}$ have two greatest lower bounds $\boldsymbol{\ell}$ and $\boldsymbol{\ell}'$, then they must be equal:
From $\boldsymbol{\ell} \trianglelefteq \mathbf{s},\ \mathbf{t}$ and $\boldsymbol{\ell}' \trianglelefteq \mathbf{s},\ \mathbf{t}$ we infer $\widehat{\boldsymbol{\ell}} \trianglelefteq \widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}}$ and $\widehat{\boldsymbol{\ell}'} \trianglelefteq \widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}}$ by Theorem 12. Since furthermore $\boldsymbol{\ell}$ and $\boldsymbol{\ell}'$ are greatest lower bounds, $\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}} \trianglelefteq \widehat{\boldsymbol{\ell}}$ and $\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}} \trianglelefteq \widehat{\boldsymbol{\ell}'}$. Thus $\widehat{\boldsymbol{\ell}} = \widehat{\boldsymbol{\ell}'}$. By Theorem 11, the only way in which $\boldsymbol{\ell} \neq \boldsymbol{\ell}'$ is that

$$\boldsymbol{\ell} = (\widehat{\boldsymbol{\ell}})_+, \quad \boldsymbol{\ell}' = (\widehat{\boldsymbol{\ell}})^+ \qquad \mathbf{or} \qquad \boldsymbol{\ell} = (\widehat{\boldsymbol{\ell}})^+, \quad \boldsymbol{\ell}' = (\widehat{\boldsymbol{\ell}})_+$$

but then either $\boldsymbol{\ell} \trianglelefteq \boldsymbol{\ell}'$ or $\boldsymbol{\ell}' \trianglelefteq \boldsymbol{\ell}$, contradicting their both being greatest lower bounds. So $\boldsymbol{\ell} = \boldsymbol{\ell}'$. □

We can therefore compute $\mathbf{s} \wedge \mathbf{t}$ and $\mathbf{s} \vee \mathbf{t}$ with the following recursive algorithms, where the expansion used is chosen from among the lower and upper expansions:

$$\mathbf{s} \wedge \mathbf{t} \quad = \quad \begin{cases} \mathbf{s} & \text{if} \quad \mathbf{s} \trianglelefteq \mathbf{t} \\ \mathbf{t} & \text{if} \quad \mathbf{t} \trianglelefteq \mathbf{s} \\ \text{the greatest expansion of } \ \widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}} & \\ \text{that is also a lower bound for } \mathbf{s} \text{ and } \mathbf{t} & \text{otherwise;} \end{cases}$$

$$\mathbf{s} \vee \mathbf{t} \quad = \quad \begin{cases} \mathbf{t} & \text{if} \quad \mathbf{s} \trianglelefteq \mathbf{t} \\ \mathbf{s} & \text{if} \quad \mathbf{t} \trianglelefteq \mathbf{s} \\ \text{the least expansion of } \ \widehat{\mathbf{s}} \vee \widehat{\mathbf{t}} & \\ \text{that is also an upper bound for } \mathbf{s} \text{ and } \mathbf{t} & \text{otherwise.} \end{cases}$$

It may help to think of these algorithms as similar to the recursion used by the Huffman algorithm: for solving the problem of size n, they combine the two deepest leaves (as a contraction), and recursively solve a problem of size n–1; finally, they find the best possible (upper or lower) expansion. The following table of nontrivial examples gives an appreciation for glbs and lubs.

$\mathbf{s}$	$\mathbf{t}$	$\mathbf{s} \wedge \mathbf{t}$	$\mathbf{s} \vee \mathbf{t}$
(1 2 4 5 5 5 5 5 5 5)	(2 2 2 3 4 5 6 7 7)	(2 2 2 4 4 5 5 5 5)	(1 2 4 4 4 5 6 7 7)
(1 2 4 5 5 5 5 5 5 5)	(2 2 3 3 3 4 5 6 6)	(2 2 3 3 3 5 5 5 5)	(1 2 4 4 5 5 5 6 6)
(1 3 4 4 4 4 4 5 5)	(2 2 3 3 3 4 5 6 6)	(2 2 3 3 4 4 4 5 5)	(1 3 3 4 4 4 5 6 6)
(1 4 4 4 4 4 4 4 4)	(2 2 3 3 3 4 5 6 6)	(2 2 3 4 4 4 4 4 4)	(1 3 3 4 4 4 5 6 6)
(1 4 4 4 4 4 4 4 4)	(2 2 2 3 5 5 5 6 6)	(2 2 3 4 4 4 4 4 4)	(1 3 3 3 5 5 5 6 6)

The final example in this table is expanded to show how it can be derived with the recursive algorithms above. Comparing with Figure 5 may help in studying this expansion.

n	$\mathbf{s}$	$\mathbf{t}$	$\mathbf{s} \wedge \mathbf{t}$	$\mathbf{s} \vee \mathbf{t}$
9	(1 4 4 4 4 4 4 4 4)	(2 2 2 3 5 5 5 6 6)	(2 2 3 4 4 4 4 4 4)	(1 3 3 3 5 5 5 6 6)
	$\uparrow$ lower expansion	$\uparrow$ upper expansion	$\uparrow$ lower expansion	$\uparrow$ upper expansion
8	(1 3 4 4 4 4 4 4)	(2 2 2 3 5 5 5 5)	(2 2 3 3 4 4 4 4)	(1 3 3 3 5 5 5 5)
	$\uparrow$ lower expansion	$\uparrow$ lower expansion	$\uparrow$ lower expansion	$\uparrow$ lower expansion
7	(1 3 3 4 4 4 4)	(2 2 2 3 4 5 5)	(2 2 2 4 4 4 4)	(1 3 3 3 4 5 5)
	$\uparrow$ lower expansion	$\uparrow$ upper expansion	$\uparrow$ lower expansion	$\uparrow$ upper expansion
6	(1 3 3 3 4 4)	(2 2 2 3 4 4)	(2 2 2 3 4 4)	(1 3 3 3 4 4)

Note the final pairs of entries in $\mathbf{s}$ and $\mathbf{t}$ are the same as the final pairs of entries in $\mathbf{s} \wedge \mathbf{t}$ and $\mathbf{s} \vee \mathbf{t}$, and the suffix lengths of $\mathbf{s}$ and $\mathbf{t}$ are never shorter than those of $\mathbf{s} \wedge \mathbf{t}$ and $\mathbf{s} \vee \mathbf{t}$. These observations suggest the following theorem. It is provable via majorization, but we sketch a direct proof.

Theorem 14 *If* $\mathbf{s}$ *and* $\mathbf{t}$ *are path-length sequences of length* n*, then:*

$$\mathbf{s} \vee \mathbf{t} \;=\; \begin{cases} (\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})^+ & \textit{if} \quad \mathbf{s} = (\widehat{\mathbf{s}})^+ \quad \textit{and} \quad \mathbf{t} = (\widehat{\mathbf{t}})^+ \\ (\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})_+ & \textit{if} \quad \mathbf{s} = (\widehat{\mathbf{s}})_+ \quad \textit{and} \quad \mathbf{t} = (\widehat{\mathbf{t}})_+ \end{cases}$$

$$\mathbf{s} \wedge \mathbf{t} \;=\; \begin{cases} (\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}})^+ & \textit{if} \quad \mathbf{s} = (\widehat{\mathbf{s}})^+ \quad \textit{and} \quad \mathbf{t} = (\widehat{\mathbf{t}})^+ \\ (\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}})_+ & \textit{if} \quad \mathbf{s} = (\widehat{\mathbf{s}})_+ \quad \textit{and} \quad \mathbf{t} = (\widehat{\mathbf{t}})_+ \end{cases}$$

Also, if either $\mathbf{s} = (\widehat{\mathbf{s}})^+$ *and* $\mathbf{t} = (\widehat{\mathbf{t}})_+$*, or* $\mathbf{s} = (\widehat{\mathbf{s}})_+$ *and* $\mathbf{t} = (\widehat{\mathbf{t}})^+$*,*
then either $\mathbf{s} \wedge \mathbf{t} = (\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}})^+$ *and* $\mathbf{s} \vee \mathbf{t} = (\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})_+$*, or* $\mathbf{s} \wedge \mathbf{t} = (\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}})_+$ *and* $\mathbf{s} \vee \mathbf{t} = (\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})^+$.
Furthermore: if the final pairs of entries of $\mathbf{s}$ *and* $\mathbf{t}$ *are* $(p\ p)$ *and* $(q\ q)$*, where* $p \leq q$*, then the final pairs of entries of* $\mathbf{s} \wedge \mathbf{t}$ *and* $\mathbf{s} \vee \mathbf{t}$ *are respectively* $(p\ p)$ *and* $(q\ q)$*.*
Moreover: the suffix lengths of $\mathbf{s}$ *and* $\mathbf{t}$ *are at least as long as those of* $(\mathbf{s} \wedge \mathbf{t})$ *and* $(\mathbf{s} \vee \mathbf{t})$*.*

Proof
These properties follow by induction on n. For the basis, they all hold trivially when $n \leq 6$, since then the imbalance lattice is a total order and $\{\,\mathbf{s},\ \mathbf{t}\,\} = \{\,\mathbf{s} \vee \mathbf{t},\ \mathbf{s} \wedge \mathbf{t}\,\}$, and the final two entries of any path-length sequence are a pair by Theorem 2. For the induction step, we can write

$$\mathbf{s} \;=\; (\,\cdots\,(p{-}i)\;\overbrace{p\;\;\cdot\;\;\cdot\;\;\cdot\;\;p}^{2h}\,) \qquad \mathbf{t} \;=\; (\,\cdots\,(q{-}j)\;\overbrace{q\;\;\cdot\;\;\cdot\;\;\cdot\;\;q}^{2k}\,)$$

$$\widehat{\mathbf{s}} \;=\; (\,\cdots\,(p{-}i)\,(p{-}1)\;\overbrace{p\,\cdots\,p}^{2h-2}\,) \qquad \widehat{\mathbf{t}} \;=\; (\,\cdots\,(q{-}j)\,(q{-}1)\;\overbrace{q\,\cdots\,q}^{2k-2}\,)$$

where $i, j, h, k > 0$, and we assume with no loss of generality that $p \leq q$. There are four cases to consider, depending on the the suffix lengths $2h$ of $\mathbf{s}$ and $2k$ of $\mathbf{t}$. In the first, $h = 1$ and $k = 1$, i.e., $\mathbf{s} = (\widehat{\mathbf{s}})^+$ and $\mathbf{t} = (\widehat{\mathbf{t}})^+$. Then $i = 1$ and $j = 1$ by Theorem 2. By induction $(\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}})$ and $(\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})$ have respective final pairs $((p{-}1)\,(p{-}1))$ and $((q{-}1)\,(q{-}1))$, and have suffix lengths not exceeding those of $\widehat{\mathbf{s}}$ and $\widehat{\mathbf{t}}$. Now, by Theorem 11 $(\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}})^+ \trianglelefteq (\widehat{\mathbf{s}})^+$ and $(\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}})^+ \trianglelefteq (\widehat{\mathbf{t}})^+$. But $(\widehat{\mathbf{s}})^+ = \mathbf{s}$ and $(\widehat{\mathbf{t}})^+ = \mathbf{t}$, so the recursive algorithm above asserts $(\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}})^+ = \mathbf{s} \wedge \mathbf{t}$. Thus the final pair of $\mathbf{s} \wedge \mathbf{t}$ will be $(p\ p)$, and it has suffix length 2. Similarly $\mathbf{s} \vee \mathbf{t} = (\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})^+$ because $\mathbf{s} \vee \mathbf{t} \in \{\,(\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})_+,\ (\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})^+\,\}$, and choosing $(\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})_+$ gives a contradiction: if $\mathbf{s} = (\widehat{\mathbf{s}})^+ \trianglelefteq (\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})_+$ and $\mathbf{t} = (\widehat{\mathbf{t}})^+ \trianglelefteq (\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})_+$ then $\widehat{\mathbf{s}} = \widehat{(\widehat{\mathbf{s}})^+} \trianglelefteq \widehat{(\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})_+} \neq \widehat{(\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})^+} = \widehat{\mathbf{s}} \vee \widehat{\mathbf{t}}$ and $\widehat{\mathbf{t}} = \widehat{(\widehat{\mathbf{t}})^+} \trianglelefteq \widehat{(\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})_+} \neq \widehat{(\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})^+} = \widehat{\mathbf{s}} \vee \widehat{\mathbf{t}}$ so the lub of $\widehat{\mathbf{s}}$ and $\widehat{\mathbf{t}}$ is not $\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}}$, a contradiction. Again the final pair of $\mathbf{s} \vee \mathbf{t}$ will be $(q\ q)$, with suffix length 2. The other three cases, where $h > 1$ and/or $k > 1$, are similar. $\square$

4 Properties of Weighted Path-Length over the Lattice

In Huffman coding we are given a positive descending weight sequence $\mathbf{w} = (\, w_1\, w_2\, \cdots\, w_n\,)$ and are asked to find a binary tree path-length sequence $\boldsymbol{\ell} = (\, \ell_1\, \ell_2\, \cdots\, \ell_n\,)$ minimizing the weighted path-length

$$g_{\mathbf{w}}(\boldsymbol{\ell}) \quad = \quad \sum_{i=1}^{n} w_i\, \ell_i .$$

In this section we first show that $g_{\mathbf{w}}$ is not monotone over the lattice of trees. However, we show that it is submodular, which helps explain why efficient algorithms for finding optimal trees are possible at all.

4.1 Non-monotonicity of Weighted Path-Length over the Lattice

The results of the previous sections can now be applied to Huffman coding. Consider the weight sequence $\mathbf{w} = (\, 10\, 9\, 8\, 7\, 3\, 2\, 1\,)$. This sequence yields the total weighted path-length for each of the possible tree path-length sequences for $n = 7$ presented graphically on the balancing exchange diagram in Figure 10. The unique optimal (Huffman) sequence for this is $(\, 2\, 2\, 2\, 3\, 4\, 5\, 5\,)$, with weighted path-length 102.

Path-length Sequence	Weighted Path-length
1234566	113
1235555	117
1244455	115
1333455	109
1334444	113
2223455	102
2224444	106
2233344	104
2333333	110

```
      1234566
     /   |113\
    /    |   \\
   /     |    \\
  /   1235555  )\
 /    /  |117\/  \
(    /   |   /\   \
 \  /    |  /  \   \
  \(  1244455   \   \
   \\/   |115\   \   \
    \\   |   \\   \   \
   / \\  |    \\   \   \
  /   1333455  )\   \   \
 /   /   |109\/  \   \   )
(   /    |   /\   \   ) /
 \ /     |  /  \   \ / /
  \   1334444   2223455
 / \ /  |113  /   /102
(   \/   |    /   /
 \  /\   |   /   /
  \(  2224444   /
   \\    |106  /
    \\   |    /
     \\  |   /
      2233344
         |104
         |
         |
      2333333
          110
```

Figure 10: Total weighted path-lengths when $\mathbf{w} = (\, 10\, 9\, 8\, 7\, 3\, 2\, 1\,)$

This small example shows that $g_{\mathbf{w}}$ is not monotone on the lattice (far from it). Thus, an algorithm for minimizing $g_{\mathbf{w}}(\boldsymbol{\ell})$ cannot simply move naively up or down the lattice, etc.

However, the example does leave open the possibility that a greedy algorithm could find the optimal code. It appears there might be sufficient connectivity among points in the lattice that the optimum path-length sequence can be reached by a sequence of decreasing-cost steps from any point. A descending positive weight sequence

$$(\cdots \quad w_1 \quad \cdots \quad w_2 \quad w_3 \quad \cdots)$$

for which the tree path-lengths are

$$(\cdots \quad (\ell-1) \quad \cdots \quad (\ell+1) \quad (\ell+1) \quad \cdots)$$

can be improved by a minimal balancing exchange if the tree path-lengths

$$(\cdots \quad \ell \quad \cdots \quad \ell \quad \ell \quad \cdots)$$

decrease the weighted path-length of the tree. Specifically, the balancing exchange yields a net improvement when $\ell\, w_1 + \ell\, w_2 + \ell\, w_3 \le (\ell-1)\, w_1 + (\ell+1)\, w_2 + (\ell+1)\, w_3$, or

$$w_1 \quad \le \quad w_2 + w_3.$$

Because there is no minimal balancing exchange from the sequence (1 4 4 4 4 4 4 4 4), for example, it is insufficient to consider only minimal balancing exchanges. So, more generally, given a sequence of $k \ge 3$ weights

$$(\cdots \quad w_1 \quad w_2 \quad w_3 \quad \cdots\cdots\cdots \quad w_{k-2} \quad w_{k-1} \quad w_k \quad \cdots)$$

and the path-length sequence

$$(\cdots \quad \ell_1 \quad \ell_2 \quad \ell_3 \quad \cdots\cdots\cdots \quad \ell_{k-2} \quad (\ell_{k-1}+1) \quad (\ell_{k-1}+1) \quad \cdots)$$

then the balancing exchange to

$$(\cdots \quad (\ell_1+1) \quad (\ell_1+1) \quad \ell_2 \quad \cdots\cdots\cdots \quad \ell_{k-3} \quad \ell_{k-2} \quad \ell_{k-1} \quad \cdots)$$

yields a net improvement when

$$(w_1 + w_2) - (w_{k-1} + w_k) \quad \le \quad w_2(\ell_2-\ell_1) + w_3(\ell_3-\ell_2) + \cdots + w_{k-1}(\ell_{k-1}-\ell_{k-2}).$$

Will a greedy algorithm following this exchange step always find its way to an optimal solution?

The more detailed example shown in Figure 11 proves that the answer to this question is no, and that greedy search is unfortunately not optimal. For this problem the sequence (2 2 3 3 4 4 4 5 5) with cost 1298 is a local minimum: each of the 7 sequences reachable from it by imbalancing exchanges, and each of the 3 sequences reachable from it by balancing exchanges, have greater weighted path length. The diagram shows only the transitive reduction of the imbalance lattice, omitting many balancing exchanges (because they would clutter the picture), but it conveys the general situation for larger Huffman coding problems. It shows that simple hill-climbing along ternary exchanges is not guaranteed to find the optimum sequence.

4.2 Submodularity of Weighted Path-Length over the Lattice

In this section we show that weighted path-length on the imbalance lattice of trees (or a logarithmic variant on the majorization lattice of densities) gives a significant example of a submodular function. This is interesting also because most work on submodular functions assumes the lattice is the lattice of subsets of a given set, the case originally emphasized by Edmonds [1].

Definition 8 *A real-valued function $f : \mathcal{L} \to \Re$ defined on a lattice $\langle \mathcal{L}, \sqsubseteq, \sqcap, \sqcup \rangle$ is* **submodular** *if*

$$f(x \sqcap y) + f(x \sqcup y) \quad \le \quad f(x) + f(y)$$

for all $x, y \in \mathcal{L}$.

Path-Length Sequence	Weighted Path-length
123456788	1329
123457777	1340
123466677	1324
123555677	1351
123556666	1362
124445677	1330
124446666	1341
124455566	1325
124555555	1386
133345677	1281
133346666	1292
133355566	1276
133444566	1303
133445555	1314
134444455	1348
144444444	1433
222345677	1302
222346666	1313
222355566	1297
222444566	1324
222445555	1335
223334566	1303
223335555	1314
223344455	1298
223444444	1359
233333455	1349
233334444	1360
333333344	1510

```
123456788
    |1329
    |
    |
123457777
    |1340
    |
    |
123466677
    |1324
    |
    |
123555677
    |1351\
    |     \
    |      \
123556666   124445677
    |1362  /    |1330
    |     /     |
    |    /      |
124446666   133345677
    |1341\      |1281\
    |     \     |     \
    |      \    |      \
124455566   133346666   222345677
    |1325\      |1292\      |1302
    |     \     |     \     |
    |      \    |      \    |
124555555   133355566   222346666
     1386       |1276\      |1313
      \         |     \     |
       \        |      \    |
        \   133444566   222355566
         \      |1303\      |1297
          \     |     \     |
           \    |      \    |
            133445555   222444566
                |1314\      |1324\
                |     \     |     \
                |      \    |      \
            134444455   222445555   223334566
                |1348       |1335  /     1303
                |   \       |     /
                |    \      |    /
            144444444 \ 222335555
                 1433  \    |1314
                  \     \   |
                   \     \  |
                    \   223344455
                     \      |1298\
                      \     |     \
                       \    |      \
                        223444444   233333455
                            |1359  /     1359
                            |     /
                            |    /
                        233334444
                            |1360
                            |
                            |
                        333333344
                             1510
```

The transitively reduced imbalance lattice for $n = 9$ showing, for each path-length sequence ℓ, the Huffman code cost $g_{\mathbf{w}}(\ell)$ with $\mathbf{w} = (\ 189\ 95\ 73\ 71\ 23\ 21\ 18\ 9\ 1\ )$. Because the graph shows only the transitive reduction of the lattice, it omits some edges corresponding to balancing exchanges. Thus for example the optimal code 133355566, *with cost 1276, is adjacent not only to* 124455566, 133346666, 133444566, 222355566 *as drawn, but also to* 123555677, 123556666, 133345677, 133445555, 223334566, 223335555.

Figure 11: Cost comparison among Huffman codes for weights $\mathbf{w} = (\ 189\ 95\ 73\ 71\ 23\ 21\ 18\ 9\ 1\ )$.

Lawler [13] has remarked:

> If a discrete optimization problem can be solved efficiently, it is quite likely that submodularity is responsible. In recent years there has been a growing appreciation of the fact that submodularity plays a pivotal role in discrete optimization, not unlike that of convexity in continuous optimization.

The close relationship between submodularity and convexity was neatly identified by Lovász [14].

Define the function $G_{\mathbf{w}}$ on the majorization lattice of exponentiated path-length sequences by

$$G_{\mathbf{w}}(\mathbf{x}) \quad = \quad g_{\mathbf{w}}(-\log_2(\mathbf{x})) \quad = \quad -\sum_i w_i \; \log_2(x_i).$$

Theorem 15 *Assuming* $\mathbf{w}$ *is a descending positive sequence of length* n*,* $G_{\mathbf{w}}$ *is submodular on the majorization lattice of nowhere-zero densities. That is, for all strictly positive density sequences* $\mathbf{x}$*,* $\mathbf{y}$ *of length* n*,*

$$G_{\mathbf{w}}(\mathbf{x} \sqcap \mathbf{y}) \; + \; G_{\mathbf{w}}(\mathbf{x} \sqcup \mathbf{y}) \quad \leq \quad G_{\mathbf{w}}(\mathbf{x}) \; + \; G_{\mathbf{w}}(\mathbf{y}).$$

Proof

By induction on n. For $n = 1$, the inequality is satisfied with equality. Let a_n and b_n be the n-th entries of $(\mathbf{x} \sqcap \mathbf{y})$ and $(\mathbf{x} \sqcup \mathbf{y})$, respectively. The theorem follows by induction if we can show

$$w_n \cdot (-\log_2(a_n)) \; + \; w_n \cdot (-\log_2(b_n)) \quad \leq \quad w_n \cdot (-\log_2(x_n)) \; + \; w_n \cdot (-\log_2(y_n)).$$

Recalling that

$$\mathbf{x} \sqcap \mathbf{y} \quad = \quad D\left((S\,\mathbf{x})\, \mathbf{min}^{+}\, (S\,\mathbf{y})\right), \qquad \mathbf{x} \sqcup \mathbf{y} \quad = \quad D\left((S\,\mathbf{x})\, \mathbf{max}^{+}\, (S\,\mathbf{y})\right),$$

there are four cases, depending on $\mathbf{X} = S\,\mathbf{x}$ and $\mathbf{Y} = S\,\mathbf{y}$, and specifically on the final values

$$X_{n-1} \; = \; \sum_{i=1}^{n-1} x_i, \qquad X_n \; = \; \sum_{i=1}^{n} x_i, \qquad Y_{n-1} \; = \; \sum_{i=1}^{n-1} y_i, \qquad Y_n \; = \; \sum_{i=1}^{n} y_i$$

as follows:

1. if $X_{n-1} \leq Y_{n-1}$ and $X_n \leq Y_n$, then $a_n = x_n$, $b_n = y_n$.
2. if $X_{n-1} \geq Y_{n-1}$ and $X_n \geq Y_n$, then $a_n = y_n$, $b_n = x_n$.
3. if $X_{n-1} \leq Y_{n-1}$ and $X_n \geq Y_n$, then $x_n \geq y_n$, $a_n = Y_n - X_{n-1} = y_n + \epsilon$, $b_n = X_n - Y_{n-1} = x_n - \epsilon$, where $\epsilon = (Y_{n-1} - X_{n-1}) \geq 0$ and $\epsilon \leq x_n - y_n = (x_n \max y_n) - (x_n \min y_n)$.
4. if $X_{n-1} \geq Y_{n-1}$ and $X_n \leq Y_n$, then $y_n \geq x_n$, $a_n = X_n - Y_{n-1} = x_n + \epsilon$, $b_n = Y_n - X_{n-1} = y_n - \epsilon$, where $\epsilon = (X_{n-1} - Y_{n-1}) \geq 0$ and $\epsilon \leq y_n - x_n = (x_n \max y_n) - (x_n \min y_n)$.

Each case satisfies $w_n \cdot (-\log_2(a_n)) + w_n \cdot (-\log_2(b_n)) \leq w_n \cdot (-\log_2(x_n)) + w_n \cdot (-\log_2(y_n))$ as needed; the first two cases satisfy it with equality, and in the last two we have

$$a_n \quad = \quad (x_n \min y_n) + \epsilon, \qquad b_n \quad = \quad (x_n \max y_n) - \epsilon$$

but then assuming $0 < x_n,\, y_n \leq 1$,

$$\log_2(a_n) + \log_2(b_n) \quad = \quad \log_2(a_n\, b_n) \quad = \quad \log_2(x_n\, y_n \; + \; \epsilon\, c) \quad \geq \quad \log_2(x_n) + \log_2(y_n)$$

where $c = ((x_n \max y_n) - (x_n \min y_n) - \epsilon) \geq 0$ and multiplying by $-w_n$ gives the theorem. □

Theorem 16 *Assuming* $\mathbf{w}$ *is a descending positive sequence of length* n*,* $g_{\mathbf{w}}$ *is submodular on the imbalance lattice. That is, for all path-length sequences* $\mathbf{s}$*,* $\mathbf{t}$ *of length* n*,*

$$g_{\mathbf{w}}(\mathbf{s} \wedge \mathbf{t}) \;+\; g_{\mathbf{w}}(\mathbf{s} \vee \mathbf{t}) \quad\leq\quad g_{\mathbf{w}}(\mathbf{s}) \;+\; g_{\mathbf{w}}(\mathbf{t}).$$

Proof

Also by induction on n. The theorem holds with equality for $n \leq 6$, since then the lattice of path-length sequences is totally ordered. We sketch the induction step from $n-1$ to n, assuming $\Delta g_{\mathbf{w}}(\widehat{\mathbf{s}}, \widehat{\mathbf{t}}) \;=\; \big(g_{\mathbf{w}}(\widehat{\mathbf{s}}) + g_{\mathbf{w}}(\widehat{\mathbf{t}})\big) \;-\; \big(g_{\mathbf{w}}(\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}}) + g_{\mathbf{w}}(\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})\big) \;\geq\; 0$ (where $g_{\mathbf{w}}$, when applied to sequences of length $n-1$, uses only the first $n-1$ entries of $\mathbf{w}$), and showing that $\Delta g_{\mathbf{w}}(\mathbf{s}, \mathbf{t}) \geq 0$.

Recall that $2k$ is the suffix length of the path-length sequence $\boldsymbol{\ell} = (\;\cdots\;(q-j)\;\overbrace{q\;\cdots\;q}^{2k}\;)$, and j is its suffix increment. The suffix increment of $\widehat{\boldsymbol{\ell}}$ is 1 when $\boldsymbol{\ell} = (\widehat{\boldsymbol{\ell}})_{+}$, so:

$$g_{\mathbf{w}}(\boldsymbol{\ell}) \quad=\quad \begin{cases} g_{\mathbf{w}}(\widehat{\boldsymbol{\ell}}) \;+\; (w_{n-1} \quad\;\; +\; q \cdot w_n) & \text{if } \boldsymbol{\ell} = (\widehat{\boldsymbol{\ell}})^{+} \quad \text{(i.e., } k = 1) \\ g_{\mathbf{w}}(\widehat{\boldsymbol{\ell}}) \;+\; (w_{n-2k+1} \;+\; q \cdot w_n) & \text{if } \boldsymbol{\ell} = (\widehat{\boldsymbol{\ell}})_{+} \quad \text{(i.e., } k > 1). \end{cases}$$

Thus $g_{\mathbf{w}}(\mathbf{s}) > g_{\mathbf{w}}(\widehat{\mathbf{s}})$ and $g_{\mathbf{w}}(\mathbf{t}) > g_{\mathbf{w}}(\widehat{\mathbf{t}})$ in all cases.

However, it can happen that $g_{\mathbf{w}}(\mathbf{s} \wedge \mathbf{t}) < g_{\mathbf{w}}(\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}})$ or $g_{\mathbf{w}}(\mathbf{s} \vee \mathbf{t}) < g_{\mathbf{w}}(\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})$ because it is possible that $\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}} \neq \widehat{\mathbf{s} \wedge \mathbf{t}}$ or $\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}} \neq \widehat{\mathbf{s} \vee \mathbf{t}}$. Specifically, it is possible that $\mathbf{s} \wedge \mathbf{t} = (\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}})_{+}$ but $\mathbf{s} \wedge \mathbf{t}$ has suffix increment $j > 1$. Then $\mathbf{s} \wedge \mathbf{t} \;=\; (\;\cdots\;(q-j)\;(q-j)\;\overbrace{q\;\cdots\;q}^{2k}\;)$, $\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}} \;=\; (\;\cdots\;(q-j-1)\;\overbrace{q\;\cdots\;q}^{2k}\;)$, so

$$g_{\mathbf{w}}(\mathbf{s} \wedge \mathbf{t}) \quad=\quad g_{\mathbf{w}}(\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}}) \;+\; (w_{n-2k+1} \;-\; j \cdot w_{n-2k} \;+\; q \cdot w_n)$$

and the parenthesized expression can be negative.

From Theorem 14, the final pairs of entries of $\mathbf{s}$ and $\mathbf{t}$ are always the same as the final pairs of entries of $\mathbf{s} \wedge \mathbf{t}$ and $\mathbf{s} \vee \mathbf{t}$, and the suffix lengths for each of $\mathbf{s}$ and $\mathbf{t}$ cannot be less than those for each of $(\mathbf{s} \wedge \mathbf{t})$ and $(\mathbf{s} \vee \mathbf{t})$. We now consider the same four cases addressed in the proof of Theorem 14.

In the case where both $\mathbf{s}$ is the upper expansion of $\widehat{\mathbf{s}}$, and $\mathbf{t}$ is the upper expansion of $\widehat{\mathbf{t}}$, then by Theorem 14 $\mathbf{s} \wedge \mathbf{t} = (\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}})^{+}$ and $\mathbf{s} \vee \mathbf{t} = (\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})^{+}$, so

$$\begin{aligned} \Delta g_{\mathbf{w}}(\mathbf{s}, \mathbf{t}) \;&=\; (g_{\mathbf{w}}(\mathbf{s}) \;+\; g_{\mathbf{w}}(\mathbf{t})) \;-\; (g_{\mathbf{w}}(\mathbf{s} \wedge \mathbf{t}) \;+\; g_{\mathbf{w}}(\mathbf{s} \vee \mathbf{t})) \\ &=\; (g_{\mathbf{w}}(\widehat{\mathbf{s}}) \;+\; g_{\mathbf{w}}(\widehat{\mathbf{t}})) \;-\; (g_{\mathbf{w}}(\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}}) \;+\; g_{\mathbf{w}}(\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})) \quad=\quad \Delta g_{\mathbf{w}}(\widehat{\mathbf{s}}, \widehat{\mathbf{t}}) \end{aligned}$$

with the analysis above for $g_{\mathbf{w}}(\boldsymbol{\ell})$ with $k = 1$. In this situation only the final pairs of entries of $\mathbf{s}$, $\mathbf{t}$ and of $\mathbf{s} \wedge \mathbf{t}$, $\mathbf{s} \vee \mathbf{t}$ can cause the two differences to be unequal, but we now know them to give the same two pairs. So in this case the theorem follows by induction.

It remains to treat the cases where $\mathbf{s}$ is the lower expansion of $\widehat{\mathbf{s}}$ or $\mathbf{t}$ is the lower expansion of $\widehat{\mathbf{t}}$. In these cases it can happen that $g_{\mathbf{w}}(\mathbf{s} \wedge \mathbf{t}) < g_{\mathbf{w}}(\mathbf{s} \wedge \mathbf{t})$ or $g_{\mathbf{w}}(\mathbf{s} \vee \mathbf{t}) < g_{\mathbf{w}}(\mathbf{s} \vee \mathbf{t})$ as noted above.

In the case where either $\mathbf{s}$ is the lower expansion of $\widehat{\mathbf{s}}$ or $\mathbf{t}$ is the lower expansion of $\widehat{\mathbf{t}}$, but not both, then by Theorem 14, either $\mathbf{s} \wedge \mathbf{t} \;=\; (\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}})_{+}$ and $\mathbf{s} \vee \mathbf{t} \;=\; (\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})^{+}$, or $\mathbf{s} \wedge \mathbf{t} \;=\; (\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}})^{+}$ and $\mathbf{s} \vee \mathbf{t} \;=\; (\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})_{+}$. The lower expansions among these two cannot yield as large a $g_{\mathbf{w}}$ increase as the lower expansions giving $\mathbf{s}$ and $\mathbf{t}$, because they expand higher-indexed positions (their suffix lengths are never longer), and the suffix increment of $\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}}$ or $\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}}$ can be greater than 1. Therefore $\Delta g_{\mathbf{w}}(\mathbf{s}, \mathbf{t}) \;\geq\; \Delta g_{\mathbf{w}}(\widehat{\mathbf{s}}, \widehat{\mathbf{t}})$.

In the final case where both $\mathbf{s}$ is the lower expansion of $\widehat{\mathbf{s}}$ and $\mathbf{t}$ is the lower expansion of $\widehat{\mathbf{t}}$, then $\mathbf{s} \wedge \mathbf{t} \;=\; (\widehat{\mathbf{s}} \wedge \widehat{\mathbf{t}})_{+}$ and $\mathbf{s} \vee \mathbf{t} \;=\; (\widehat{\mathbf{s}} \vee \widehat{\mathbf{t}})_{+}$ (Theorem 14), and because the lower expansions giving $\mathbf{s} \wedge \mathbf{t}$ and $\mathbf{s} \vee \mathbf{t}$ cannot yield as large a $g_{\mathbf{w}}$ increase as the lower expansions giving $\mathbf{s}$ and $\mathbf{t}$, again $\Delta g_{\mathbf{w}}(\mathbf{s}, \mathbf{t}) \;\geq\; \Delta g_{\mathbf{w}}(\widehat{\mathbf{s}}, \widehat{\mathbf{t}})$. □

5 Future Work

The results here can be used to gain further insight about the Huffman coding problem, and perhaps also about the applications of lattice concepts in coding.

5.1 Theoretical Extensions

Almost all of the theorems proved here admit interesting extensions and/or special cases. For example, a direct corollary of Theorem 16 (using $\mathbf{w} = (\,1 \cdots 1\,)$) is that the function mapping a path-length sequence to its level of balance is submodular on the imbalance lattice.

It is not hard to show that an optimal Huffman tree of size n is an expansion (either the upper or the lower expansion) of a Huffman tree of size $(n-1)$. Specifically, upper expansion gives an optimal tree if $(w_{n-1} + w_{n-2}) < w_{n-3}$, and lower expansion gives an optimal tree otherwise. This gives a recursive algorithm for Huffman tree construction that is much like the original, so it is not new, but it gives some perspective.

It may be possible to define $\mathbf{s} \wedge \mathbf{t}$ in terms of $-\log_2\left(2^{-\mathbf{s}} \sqcap 2^{-\mathbf{t}}\right)$ and define $\mathbf{s} \vee \mathbf{t}$ in terms of $-\log_2\left(2^{-\mathbf{s}} \sqcup 2^{-\mathbf{t}}\right)$: they are identical much of the time, and are always related by

$$\begin{array}{rcl} 2^{-\mathbf{s}\,\wedge\,\mathbf{t}} & \preceq & 2^{-\mathbf{s}} \sqcap 2^{-\mathbf{t}} \\ 2^{-\mathbf{s}} \sqcup 2^{-\mathbf{t}} & \preceq & 2^{-\mathbf{s}\,\vee\,\mathbf{t}} \end{array}$$

(because $2^{-\mathbf{s}} \sqcap 2^{-\mathbf{t}}$ and $2^{-\mathbf{s}} \sqcup 2^{-\mathbf{t}}$ are the glb and lub with respect to majorization). For example, if $\alpha = 7 - \log_2(12) \approx 3.4150375$ and $\beta = (\alpha - 1)$, the following set of examples represent the unusual cases with $n = 9$ where $-\log_2\left(2^{-\mathbf{s}} \sqcap 2^{-\mathbf{t}}\right) \neq (\mathbf{s} \wedge \mathbf{t})$ or $-\log_2\left(2^{-\mathbf{s}} \sqcup 2^{-\mathbf{t}}\right) \neq (\mathbf{s} \vee \mathbf{t})$:

$\mathbf{s}$	$\mathbf{t}$	$-\log_2\left(2^{-\mathbf{s}} \sqcap 2^{-\mathbf{t}}\right)$	$\mathbf{s} \wedge \mathbf{t}$
(1 2 4 5 5 5 5 5 5)	(1 3 3 3 4 5 6 7 7)	(1 3 3 α 5 5 5 5 5)	(1 3 3 4 4 5 5 5 5)
(1 2 4 5 5 5 5 5 5)	(2 2 2 3 4 5 6 7 7)	(2 2 2 α 5 5 5 5 5)	(2 2 2 4 4 5 5 5 5)
(1 3 4 4 4 4 4 5 5)	(2 2 2 3 4 5 6 7 7)	(2 2 β 4 4 4 4 5 5)	(2 2 3 3 4 4 4 5 5)

$\mathbf{s}$	$\mathbf{t}$	$-\log_2\left(2^{-\mathbf{s}} \sqcup 2^{-\mathbf{t}}\right)$	$\mathbf{s} \vee \mathbf{t}$
(1 4 4 4 4 4 4 4 4)	(2 2 2 3 4 5 6 7 7)	(1 4 β 3 4 5 6 7 7)	(1 3 3 3 4 5 6 7 7)
(1 4 4 4 4 4 4 4 4)	(2 2 2 3 4 6 6 6 6)	(1 4 β 3 4 6 6 6 6)	(1 3 3 3 4 6 6 6 6)
(1 4 4 4 4 4 4 4 4)	(2 2 2 3 5 5 5 6 6)	(1 4 β 3 5 5 5 6 6)	(1 3 3 3 5 5 5 6 6)
(1 4 4 4 4 4 4 4 4)	(2 2 2 4 4 4 5 6 6)	(1 4 β 4 4 4 5 6 6)	(1 3 3 4 4 4 5 6 6)
(1 4 4 4 4 4 4 4 4)	(2 2 2 4 4 5 5 5 5)	(1 4 β 4 4 5 5 5 5)	(1 3 3 4 4 5 5 5 5)

These examples suggest there may be recursive algorithms that 'round up' $-\log_2\left(2^{-\mathbf{s}} \sqcap 2^{-\mathbf{t}}\right)$ to give $\mathbf{s} \wedge \mathbf{t}$, and 'round down' $-\log_2\left(2^{-\mathbf{s}} \sqcup 2^{-\mathbf{t}}\right)$ to give $\mathbf{s} \vee \mathbf{t}$.

Majorization, we believe, can be exploited further in characterizing optimal codes. We have established that the imbalance ordering on tree path-length sequences $\boldsymbol{\ell}$ is isomorphic to the majorization ordering on exponentiated tree path-length sequences $\mathbf{x} = 2^{-\boldsymbol{\ell}}$. Thus any function that is Schur convex (i.e., 'majorization-preserving': monotone with respect to the majorization ordering) on exponentiated path-length sequences and hence monotone on the (continuous) majorization lattice, will also be monotone on the (discrete) imbalance lattice. Negative entropy is an important example of such a function. It seems possible that numerical optimization techniques for continuous problems can be brought to bear on the coding problem via majorization.

It would be interesting to extend the work here for the t-ary codes discussed in [10].

5.2 Practical Applications in Adaptive Coding

In many practical situations it is difficult or impossible to know *a priori* the weights $\mathbf{w}$ used in Huffman coding. A natural idea, which occurred independently to Faller [2] and Gallager [5], is to allow the weights to be determined dynamically, and have the Huffman code 'evolve' over time. **Dynamic Huffman coding** is the strategy of repeatedly constructing the Huffman code for the input so far, and using it in transmitting the next input symbol. Knuth presented an efficient algorithm for dynamic Huffman coding in [11], and his performance results for the algorithm show it consistently producing compression very near (though not surpassing) the compression attained with static Huffman code for the entire input.

Vitter [24, 25] then developed a dynamic Huffman algorithm that improves on Knuth's in the following way: rather than simply revise the Huffman tree after each input symbol, Vitter also finds a new Huffman tree of minimal external path length $\sum_i \ell_i$ and height $\max_i \ell_i$. With this modification Vitter was actually able to surpass the performance of static Huffman coding on several benchmarks.

A small contribution we can make is to clarify the improvement of Vitter. Basically, Vitter's algorithm differs from Knuth's in *constructing the optimal path-length sequence that is also as balanced as possible.* Note that minimizing the external path length $\sum_i \ell_i$ is identical to maximizing the level of balance we defined earlier. Since there can be more than one optimal code, and unnecessary imbalance tends to penalize the symbol currently being encoded, Vitter correctly reasoned that insisting on maximally balanced codes gives better performance.

Another contribution of the lattice perspective here is to encourage development of new adaptive coding schemes. As discussed in Section 4.1, a move between adjacent points in the lattice corresponds to minor alteration of codes, and by moving through the lattice we incrementally modify the cost of a code. Hill-climbing then gives greedy coding algorithms, and on-line hill-climbing gives adaptive coding algorithms. Although we have shown that the codes produced greedily are not Huffman codes, lattice-oriented adaptive coding algorithms may still have a role to play in some coding situations, since the Huffman notion of optimality is not really what is needed in the (currently popular and enormously important) adaptive context.

Acknowledgement

We are very grateful to Pierre Hasenfratz for insightful comments that improved this paper.

References

[1] J. Edmonds, "Submodular Functions, Matroids and Certain Polyhedra", in *Combinatorial Structures and their Applications*, R. Guy, H. Hanani, N. Saauer, J. Schonheim, eds., 69–87, Gordon and Breach, 1970.

[2] N. Faller, "An adaptive system for data compression", *Record of the 7th Asilomar Conference on Circuits, Systems, and Computers*, 593–597, 1973.

[3] S. Fujishige, *Submodular functions and optimization*, NY: North-Holland, 1991.

[4] R.G. Gallager, *Information Theory and Reliable Communications*, NY: Wiley, 1968.

[5] R.G. Gallager, "Variations on a theme by Huffman", *IEEE Trans. Information Theory* **IT-24**:6 (November 1978), 668–674.

[6] E.N. Gilbert, "Codes Based on Inaccurate Source Probabilities", *IEEE Trans. Information Theory* **IT-17**:3 (May 1971), 304–314. $g(N)$ is analyzed on p. 309.

[7] C.R. Glassey and R.M. Karp, "On the optimality of Huffman trees", *SIAM J. Appl. Math.* **31**:1 (1976), 368–378.

[8] G.H. Hardy, J.E. Littlewood, G. Polya, *Inequalities*, Cambridge University Press, 1934.

[9] D.A. Huffman, "A method for the construction of minimum redundancy codes", *Proc. IRE* **40** (1951), 1098–1101.

[10] F.K. Hwang, "Generalized Huffman Trees", *SIAM J. Appl. Math.* **37**:1 (August 1979), 124–127.

[11] D.E. Knuth, "Dynamic Huffman Coding", *J. Algorithms* **6** (1985), 163–180.

[12] E. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, 1976.

[13] E.L. Lawler, "Submodular Functions and Polymatroid Optimization", In A.H.G. Rinnooy Kan (eds.) M. O'hEigeartaigh, J.K. Lenstra, eds., *Combinatorial Optimization: Annotated Bibliographies*, 32–38, J. Wiley & Sons, 1985.

[14] L. Lovász, Submodular functions and convexity, In B. Korte (eds.) A. Bachem, M. Grötschel, eds., *Mathematical Programming: The State of the Art*, 235–257, Springer-Verlag, 1983.

[15] U. Manber, *Introduction to Algorithms*, Addison-Wesley, 1989.

[16] A.W. Marshall, I. Olkin, *Inequalities: Theory of Majorization and Its Applications*, NY: Academic Press, 1979.

[17] A. Ostrowski, "Sur quelques applications des fonctions convexes et concaves au sens de I. Schur (offert en homage à P. Montel)", *J. Math. Pures Appl.* **31** (1952), 253–292.

[18] J.M. Pallo, "Enumerating, ranking and unranking binary trees", *Computer Journal*, **29**:2 (April 1986), 171–175.

[19] J.M. Pallo, "Some properties of the rotation lattice of binary trees," *Computer Journal*, **31**:6 (Dec. 1988), 564–565.

[20] D.S. Parker, "Conditions for Optimality of the Huffman Algorithm", *SIAM J. Comput.* **9**:3 (August 1980), 470–489.

[21] D.S. Parker, P. Ram, "Greed and Majorization", November 1994. Issued as Technical Report CSD-960003, UCLA Computer Science Dept., March 1996.

[22] G.-C. Rota, "On the Foundations of Combinatory Theory I. Theory of Möbius Functions," *Z. Wahrscheinlichkeitstheorie* **2** (1964), 340–368.

[23] I. Schur, "Über eine Klasse von Mittelbildungen mit Anwendungen auf die Determinantentheorie", *Sitzungsber. Berl. Math. Ges.* **22** (1923), 9–20.

[24] J.S. Vitter, "Design and Analysis of Dynamic Huffman Codes", *J. ACM* **34**:4 (October 1987), 825–845.

[25] J.S. Vitter, "Algorithm 673: Dynamic Huffman Coding", *ACM TOMS* **15**:2 (June 1989), 158–167.