RUTCOR
RESEARCH
REPORT

ON INSTANTANEOUS CODES

Stephan Foldes [a] Navin M. Singhi [b]

RRR 44-2004, November, 2004

RUTCOR
Rutgers Center for
Operations Research
Rutgers University
640 Bartholomew Road
Piscataway, New Jersey
08854-8003
Telephone: 732-445-3804
Telefax: 732-445-5472
Email: rrr@rutcor.rutgers.edu
http://rutcor.rutgers.edu/∼rrr

[a]Institute of Mathematics, Tampere University of Technology, PL553, 33101 Tampere, Finland, e-mail: sf@tut.fi

[b]Tata Institute of Fundamental Research, School of Mathematics, Homi Bhabha Road, Colaba, Mumbai 400 005, India, e-mail: singhi@tifr.res.in

Rutcor Research Report
RRR 44-2004, November, 2004

ON INSTANTANEOUS CODES

Stephan Foldes Navin M. Singhi

Abstract. Maximal instantaneous codes are characterized by the property that they allow unique parsing of every infinite string. The sequence of codeword lengths of a maximal instantaneous code, sequenced in lexicographic order of the codewords, completely determines the code itself. Any increasing, decreasing or unimodal reordering of such a sequence again corresponds to a maximal instantaneous code. Lexicographic length sequences are characterized by a family of Kraft-type equalities.

Acknowledgement: This research was carried out while the second named author was at the Tampere University of Technology as a Nokia Visiting Fellow. The support of the Nokia Foundation is gratefully acknowledged.

1 Strings and parsing

For general definitions and background we refer to Roman [R]. For greater precision and to accommodate some generalizations we make explicit a few definitions as we understand them.

If A is any set, called the *alphabet*, then a *string* or *word* over A is a map $I \to A$ where I is a set of positive integers such that $i \in I$, $1 \leq j \leq i$ implies $j \in I$. If I is finite then the number $\mid I \mid$ of its elements is called the *length* of the string. For $I = \phi$ we have the *empty string*, denoted θ. For infinite I, $I = \{1, 2, 3, ...\}$, we speak of an *infinite string*. String $\mathbf{a}$ is a *prefix* of string $\mathbf{b}$ if $\mathbf{a}$ as a map is a restriction of $\mathbf{b}$. A string $\mathbf{a} : I \to A$ is also denoted by $(\mathrm{a}_i : i \in I)$, where $\mathrm{a}_i = \mathbf{a}(i)$, or by $(\mathrm{a}_1, \mathrm{a}_2, \mathrm{a}_3, \ldots)$ or $\mathrm{a}_1\mathrm{a}_2\mathrm{a}_3 \ldots$.

A *code* over A is a set C of finite words over A. Members of C are called *codewords*. A code C is *instantaneous* if no codeword is a prefix of another, distinct codeword. A *maximal instantaneous code* (over a given alphabet) is an instantaneous code that is not contained properly in any larger instantaneous code (over the same alphabet).

If $(\mathbf{a}_i)_{i \in I}$ is a string of finite length strings, $\mathbf{a}_i : I_i \to A$, then the concatenation $\mathbf{a} = \mathbf{a}_1\mathbf{a}_2 \ldots$ is defined to be the string $\mathbf{a}$: $I \to A$ given by

$$I = \bigcup_{i \in I} \{ \left(\sum_{\substack{k \in I \\ k < i}} \mid I_k \mid \right) + l : \ l \in I_i \}$$

and

$$\mathbf{a}(j) = \mathbf{a}_{m_j} \left(j - \sum_{\substack{k \in I \\ k < m}} \mid I_k \mid \right)$$

when

$$m_j = min\{n \in I : \sum_{k \leq n} \mid I_k \mid \geq j\}$$

We also say that $(\mathbf{a}_i)_{i \in I}$ is a *parsing* of $\mathbf{a}$. If all the $\mathbf{a}_i$ belong to a given code C, then we speak of *parsing into codewords*.

Instantaneous codes have the *unique decipherability* property, i.e. every string of finite length admits of at most one parsing into codewords of an instantaneous code [R]. This property of instantaneous codes is easily seen to hold also for the parsing of infinite strings. We note that non-instantaneous uniquely decipherable codes do not necessarily have this uniqueness property with respect for parsing infinite strings. For example, the infinite string $011 \cdots$ can be parsed in two different ways into words of the code $\{0, 01, 11\}$.

A code is said to have *bounded length* if the set of the lengths of the codewords is bounded from above by a positive integer. All finite codes have bounded length, infinite codes over a finite alphabet never do, and infinite codes over an infinite alphabet may have bounded length or not.

Theorem 1 *Let A be an arbitrary non-empty alphabet, finite or infinite, and C a code of bounded length over A. The following are equivalent:*

(i) *C is a maximal instantaneous code different from $\{\theta\}$.*

(ii) *every infinite string over A admits of a unique parsing into codewords.*

Remark. Bounded length is necessary. Consider the code
$C = \{0, 01, 001, 0001, \cdots\}$ over $A = \{0, 1\}$. Also, if A is empty, then (ii) is vacuously true, while (i) fails for the empty code.

Proof of Theorem l. Assume (i) and let $\mathbf{w}= \mathrm{a}_1\mathrm{a}_2\ldots$ be an infinite string. Clearly $\theta \notin C$. Let l be an integer greater than any of the codeword lengths. The prefix $\mathrm{a}_1 \ldots \mathrm{a}_l$ of $\mathbf{w}$ cannot be a prefix of any codeword, thus, by maximality, some $\mathbf{c}_1 \in C$ is a prefix of $\mathrm{a}_1 \ldots \mathrm{a}_l$ and we have $\mathbf{w} = \mathbf{c}_1\mathbf{w}'$ for some infinite string $\mathbf{w}'$. Repeating the arguments for $\mathbf{w}'$ and so on we obtain a parsing of $\mathbf{w}$ into codewords, $\mathbf{w} = \mathbf{c}_1\mathbf{c}_2 \ldots$. If we had another such parsing, $\mathbf{w} = \mathbf{d}_1\mathbf{d}_2 \ldots$, then for the first index i such that $\mathbf{c}_i \neq \mathbf{d}_i$ one of $\mathbf{c}_i$ or $\mathbf{d}_i$ would have to be a prefix of the other, contradicting the instantaneous property. This shows uniqueness.

Assume that (ii) holds. If $C = \{\theta\}$ then no infinite string has a parsing into codewords. Therefore $C \neq \{\theta\}$ (and $\theta \notin C$).

If C is not instantaneous, then for some $\mathbf{c} \neq \mathbf{d}$ in C, $\mathbf{c}$ is a prefix of $\mathbf{d}$, i.e. $\mathbf{ce}$=$\mathbf{d}$ for same non-empty word $\mathbf{e}$. We show that assuming (ii) would lead to a contradiction. Consider the infinite string $\mathbf{w} = \mathbf{dd}\ldots$. This string has the obvious parsing into codewords using the codeword $\mathbf{d}$ alone. But the assumption of (ii) also provides a parsing of the infinite string $\mathbf{ew}$ into codewords, say $\mathbf{ew} = \mathbf{c}_1\mathbf{c_2}\ldots$. Sience $\mathbf{c}\,(\mathbf{ew}) = \mathbf{w}$, we would have for $\mathbf{w}$ the two different parsings.

$$\mathbf{w} = \mathbf{d}\ \mathbf{d}\ \mathbf{d}\ldots$$

$$\mathbf{w} = \mathbf{c}\ \mathbf{c}_1\ \mathbf{c}_2\ldots$$

Finally, suppose that $C \neq \{\theta\}$ is instantaneous but not maximal, say for some word $\mathbf{d} \notin C$ the code $C \cup \{\mathbf{d}\}$ is also instantaneous. Then $\mathbf{w} = \mathbf{d}\ \mathbf{d}\ \ldots$ must have a parsing $\mathbf{w} = \mathbf{c}_1\mathbf{c}_2\ldots$ into C-words, and one of $\mathbf{c}_1$ and $\mathbf{d}$ must be a prefix of the other. Since $\mathbf{c}_1 \in C$, this contradicts the instantaneous property of $C \cup \{\mathbf{d}\}$.

□

2 Lexicographic length sequences and unimodal sorting

In what follows we assume that the alphabet A is a two-element set, without loss of generality $A = \{0, 1\}$, where 0 and 1 have the ordinary meaning as numbers. Codes over this alphabet are called *binary codes*. Also, in what follows, by a code we shall always mean a finite code.

The *lexicographic order* on a binary code is the order in which $\mathbf{a} \leq \mathbf{b}$ if and only if

(i) $\mathbf{a}$ is a prefix of $\mathbf{b}$, or

(ii) $\mathbf{c}0$ is a prefix of $\mathbf{a}$ and $\mathbf{c}1$ is a prefix of $\mathbf{b}$, where $\mathbf{c}$ is the longest common prefix of $\mathbf{a}$ and $\mathbf{b}$.

The *lexicographic length sequence* of a finite binary code C is the sequence of lengths of the codewords taken in lexicographic order. For example, the lexicographic length sequence of the code $\{1, 00, 10, 11\}$ is 2,1,2,2.

No maximal instantaneous code can be empty. Also, the only maximal instantaneous code with only one codeword $\mathbf{a}$ must necessarily consist of the empty word $\mathbf{a} = \theta$ alone. (Because if $\mathbf{a} = \mathrm{a}_1 \dots \mathrm{a}_n$, $n \geq 1$, then letting $\mathbf{a}' = \mathrm{a}_1 \dots \mathrm{a}_{n-1}\,(1 - \mathrm{a}_n)$ would yield a larger instantaneous code $\{\mathbf{a}, \mathbf{a}'\}$.) Further, if a maximal instantaneous code C has at least two codewords, then $\theta \notin C$, and for any codeword $\mathbf{a} = \mathrm{a}_1 \dots \mathrm{a}_n \in C$ of maximum length $n \geq 1$, the word $\mathbf{a}' = \mathrm{a}_1 \dots \mathrm{a}_{n-1}(1 - \mathrm{a}_n)$ must also belong to C (because otherwise $C \cup \{\mathbf{a}'\}$ would be a larger instantaneous code). In fact one of $\mathbf{a}$ and $\mathbf{a}'$ is $\mathrm{a}_1 \dots \mathrm{a}_{n-1}0$ and the other is $\mathrm{a}_1 \dots \mathrm{a}_{n-1}1$, and the latter must be the immediate successor of the former in the lexicographic order of C. This justifies the hypothesis made in the following:

Lemma. *Let $l_1 \dots l_n$ be a sequence of integers, $n \geq 2$, such that for the first index i with $l_i = max(l_1 \dots l_n)$* we have *$i < n$ and $l_{i+1} = l_i$. Then $l_1 \dots l_n$ is the lexicographic length sequence of a binary maximal instantaneous code if and only if*

$$l_1 \dots l_{i-1}(l_i - 1)l_{i+2} \dots l_n$$

is the lexicographic length sequence of a binary maximal instantaneous code.

Proof. If $l_1 \dots l_i l_{i+1} \dots l_n$ is as stipulated in the statement, and C is a corresponding maximal instantaneous code, with the codewords of C being $\mathbf{c}_1, \dots, \mathbf{c}_i, \mathbf{c}_{i+1}, \dots, \mathbf{c}_n$ in lexicographic order, then $\mathbf{c}_i$ and $\mathbf{c}_{i+1}$ are of the same maximum length in C, say length $m \geq 1$, and $\mathbf{c}_i$ is of the form $\mathrm{a}_1 \dots \mathrm{a}_{m-1}0$ while $\mathbf{c}_{i+1} = \mathrm{a}_1 \dots \mathrm{a}_{m-1}1$. Then letting $\mathbf{c} = \mathrm{a}_1 \dots \mathrm{a}_{m-1}$, the code

$$(C \backslash \{\mathbf{c}_i, \mathbf{c}_{i+1}\}) \cup \{\mathbf{c}\}$$

is maximal instantaneous.

On the other hand, if C is a maximal instantaneous code of size n-1 whose codewords in lexicographic order are

$$\mathbf{c}_1, \ldots, \mathbf{c}_{i-1}, \mathbf{c}, \mathbf{c}_{i+1}, \ldots, \mathbf{c}_n \ ,$$

of respective lengths $l_1, \ldots, l_i, (l_i - 1), l_{i+2}, \ldots, l_n$, then the code

$$(C \backslash \{\mathbf{c}\}) \cup \{\mathbf{c}0, \mathbf{c}1\}$$

is again maximal instantaneous.

□

The Lemma provides in particular a linear-time recursive algorithm to determine if for a given sequence of integers there exists or not a maximal instantaneous code with that sequence as its lexicographic length sequence.

A sequence of integers $l_1, \ldots, l_n$, $n \geq 1$, is said to be *unimodal* if there are no $1 \leq i < j < m \leq n$ such that

$$l_i > l_j \text{ and } l_j < l_m$$

All increasing as well as all decreasing sequences are clearly unimodal.

For a sequence $l_1, \ldots, l_n$, $n \geq 1$, of integers, if σ is any permutation of $\{1, ..., n\}$, then the sequence

$$l_{\sigma(1)}, \ldots, l_{\sigma(n)}$$

is called a *sorting* of $l_1, \ldots, l_n$. Every sequence has an increasing, and also a decreasing, sorting (i.e. such that $l_{\sigma(1)} \leq \ldots \leq l_{\sigma(n)}$ or $l_{\sigma(1)} \geq \ldots \geq l_{\sigma(n)}$, respectively), and these in particular are unimodal.

Theorem 2 *Let $l_1, \ldots, l_n$ be a lexicographic length sequence of a binary maximal instantaneous code. Then every unimodal sorting $l_{\sigma(1)}, \ldots, l_{\sigma(n)}$ of $l_1, \ldots, l_n$ is also a lexicographic length sequence of some binary maximal instantaneous code.*

Proof. We give the proof for unimodal sorting, the negative unimodal case is entirely similar.

We proceed by induction an n. The case n=1 is obvious. For $n \geq 1$, and assuming the claim true for lesser values, let $l = max\,(l_1, \ldots, l_n)$ and i the first index with $l_i = l$. We know that $i < n$ and l_{i+l} is also equal to l. By the Lemma $l_1 \ldots l_{i-1}(l-1)l_{i+2} \ldots l_n$ is the lexicographic length sequence of some maximal instantaneous code. Since $l_{\sigma(1)} \ldots l_{\sigma(n)}$ is unimodal, all occurrences of l in the sequence $l_{\sigma(1)} \ldots l_{\sigma(n)}$ are consecutive, and there is a permutation τ, possibly different from τ, such that the sorting $l_{\tau(1)} \ldots l_{\tau(n)}$ is the same as

$l_{\sigma(1)} \dots l_{\sigma(n)}$ and for which $\tau(i+1) = \tau(i) + 1$ and i is the first index with $l_{\tau(i)} = l$. To show that $l_{\tau(1)} \dots l_{\tau(n)}$ is the lexicographic length sequence of same maximal instantaneous code, we use again the Lemma and consider the sequence

$$l_{\tau(1)} \dots l_{\tau(i-1)}(l-1)l_{\tau(i+2)} \dots l_{\tau(n)}.$$

This sequence is also unimodal and in fact it is a unimodal sorting of $l_1 \dots l_{i-1}(l-1)l_{i+2} \dots l_n$, so the inductive hypothesis applies.

□

Abstracting from the order of occurences, but not from the number of occurences, of the various numbers in the lexicographic length sequence of a maximal instantaneous code C, we obtain a multiset of integers. It is in fact these multisets and not the lexicographic length sequences that are the object of characterization by the well-known Kraft equality (see e.g. Roman [R] and also the next Section). The theory developed by Stott Parker and Prasad Ram [SPPR] also focuses on multisets, even though the multisets are canonically represented by monotone sequences. The length multiset provides less information on the code than the lexicographic length sequence, for example both codes

$$C = \{00, 010, 011, 1\}$$

$$K = \{0, 10, 110, 111\}$$

have the same length multiset, but they are distinguished by their differing lexicographic length sequences, which are 2,3,3,1 and 1,2,3,3, respectively. Indeed we have in general the following:

Theorem 3. *Let C and K be finite binary maximal instantaneous codes. If C and K have the same lexicographic length sequence, then C=K.*

Proof. By induction on the number n of terms in the common lexicographic length sequence $l_1 \dots l_n$ of C and K.

For $n = 1$, obviously $C = K = \{\theta\}$.

For $n \geq 2$, and assuming the claim proved for lesser values, let $l = max\,(l_1 \dots l_n)$, and let i be the first index with $l_i = l$. We know that $i < n$, $l_{i+1} = l$, and by the Lemma

$$l_1 \dots l_{i-1}(l-1)l_{i+2} \dots l_n$$

is the lexicographic length sequence of some maximal instantaneous code Q, which is uniquely determined by virtue of the induction hypothesis. The code Q has n-1 codewords; denote these, in lexicographic order, by

$$\mathbf{c}_1, \dots, \mathbf{c}_{i-1}, \mathbf{c}, \mathbf{c}_{i+2}, \dots, \mathbf{c}_n \tag{1}$$

On the other hand, let the codewords of C and K be enumerated, each in lexicographic order, respectively as

$$\mathbf{a}_1, \ldots, \mathbf{a}_{i-1}, \mathbf{a}_i, \mathbf{a}_{i+1}, \mathbf{a}_{i+2}, \ldots, \mathbf{a}_n$$

and

$$\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}, \mathbf{b}_i, \mathbf{b}_{i+1}, \mathbf{b}_{i+2}, \ldots, \mathbf{b}_n$$

By the comments preceding the Lemma, we must have, for some words $\mathbf{a}$ and $\mathbf{b}$ of length l-1,

$$\mathbf{a}_i = \mathbf{a}0,\ \mathbf{a}_{i+1} = \mathbf{a}1,\ \mathbf{b}_i = \mathbf{b}0,\ \mathbf{b}_{i+1} = \mathbf{b}1 \tag{2}$$

and both

$$\mathbf{a}_1, \ldots, \mathbf{a}_{i-1}, \mathbf{a}, \mathbf{a}_{i+2}, \ldots, \mathbf{a}_n$$

and

$$\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}, \mathbf{b}, \mathbf{b}_{i+2}, \ldots, \mathbf{b}_n$$

are maximal instantaneous codes enumerated lexicographically.
The induction hypothesis implies that both of these codes must coincide with Q, whose codewords are given by (1), i.e.

$$\mathbf{a}_j = \mathbf{b}_j = \mathbf{c}_j \quad \text{for } 1 \leq j \leq i-1,\ i+2 \leq j \leq n$$

Then (2), together with $\mathbf{a} = \mathbf{b} = \mathbf{c}$, implies $\mathbf{a}_i = \mathbf{b}_i$ and $\mathbf{a}_{i+1} = \mathbf{b}_{i+1}$, i.e. C=K.

□

The above proof also provides a recursive algorithm to reconstruct a maximal instantaneous code from its lexicographic length sequence.

3 Kraft type equalities for segments of length sequences

The classical characterization of length multisets of maximal instantaneous codes by the Kraft equality

$$\sum \frac{1}{2^l} = 1$$

where in the summation there is a term $1/2^l$ for each occurrence of l in the multiset, provides only a necessary but not a sufficient condition for an ordered sequence to be a lexicographic length sequence of a maximal instantaneous code. For example, the sequence 2,1,2 is not a lexicographic length sequence, even though the corresponding multiset satisfies the Kraft equality (and thus there are maximal instantaneous codes with this length multiset).

We propose an ordered refinement of the Kraft condition. For this we need the following:

Key Definitions. Let $l_1, \ldots, l_n$, $n \geq 1$, be a sequence of non-negative integers. With respect to this sequence, a segment of integers $S = [i, j] = \{t : i \leq t \leq j\}$, $1 \leq i \leq j \leq n$, is said to be *full* if for every other segment of integers Q properly containing S, $S \subset Q \subseteq [1, n]$, we have

$$min_{t \in S} l_t > min_{t \in Q} l_t$$

In other words, to say that $S = [i, j]$ is full means that

(i) if $1 < i$ then $min(l_i \ldots l_j) > l_{i-1}$, and

(ii) if $j < n$ then $min(l_i \ldots l_j) > l_{j+1}$.

Write l^*_{i-1} for l_{i-1} if $1 < i$ else let $l^*_{i-1} = 0$, write l^*_{j+1} for l_{j+1} if $j < n$ else let $l^*_{j+1} = 0$, and call $max(l^*_{i-1}, l^*_{j+1})$ the largest term near S. The extreme example of a full segment is S=[1,n], here the *largest term near* S is 0.

Theorem 4. *A sequence $l_1 \ldots l_n$ of non-negative integers is the lexicographic length sequence of some binary maximal instantaneous code if and only if for every full $[i, j]$, $1 \leq i \leq j \leq n$, with the largest term near $[i, j]$ being denoted by m, there is some integer $k \leq 2^m$ such that*

$$\frac{1}{2^{l_i}} + \cdots + \frac{l}{2^{l_j}} = \frac{k}{2^m} \tag{3}$$

Remark. For [1,n], the largest full segment, we have $m = 0$ as previously noted, thus k and $k/2^m$ are necessarily 1, and equality (3) reduces to the Kraft equality.

Proof of Theorem 4. For the lexicographic length sequence $l_1 \ldots l_n$ of a maximal instantaneous code we prove the claimed condition by induction on n. For n=1, $l_1 = 0$, [1,1] is the only full segment, m=0, and with k=1 the equality (3) holds.

Let $n > 1$ and assume the condition true for lesser values. Let $\mathbf{c}_1, \ldots, \mathbf{c}_n$ be the codewords of a maximal instantaneous code, enumerated lexicographically, with corresponding lengths $l_1, \ldots, l_n$. With respect to this length sequence, let $[i,j]$ be a full segment, with largest term near it denoted by m. As there must be codewords starting with 0 and also codewords starting with 1 (for otherwise the single-letter word 0 and 1 could be added to the code, contradicting maximality), there must exist a $1 \leq t < n$ such that $\mathbf{c}_1, \ldots, \mathbf{c}_t$ start with 0 and $\mathbf{c}_{t+1}, \ldots, \mathbf{c}_n$ start with 1. We distinguish three cases, according to whether $t < i$, $j \leq t$, or ($i \leq t$ and $t+1 \leq j$).

Case $t < i$. Let $\mathbf{d}_{t+1}, \ldots, \mathbf{d}_n$ be the words obtained from $\mathbf{c}_{t+1}, \ldots, \mathbf{c}_n$ by removing the first letter 1. Observe also that $m > 0$. The words $\mathbf{d}_{t+1}, \ldots, \mathbf{d}_n$ are all distinct and constitute a maximal instantaneous code, and this order of enumeration is their lexicographic order, with corresponding lexicographic length sequence $l_{t+1} - 1, \ldots, l_n - 1$. Further, the segment [*i-t,j-t*] is full with respect to this length sequence, and it is not difficult to see that the largest term μ near this full segment is at most $m - 1$.
Applying the inductive hypothesis, there is an integer $K \leq 2^{\mu}$ such that

$$\frac{1}{2^{l_i - 1}} + \cdots + \frac{l}{2^{l_j - 1}} = \frac{K}{2^{\mu}}$$

i.e.

$$\frac{1}{2^{l_i}} + \cdots + \frac{l}{2^{l_j}} = \frac{K}{2^{\mu+1}} \tag{4}$$

The number $k = K2^{m-(\mu+1)}$ is an integer because $m \geq \mu + 1$. Also $k \leq 2^m$ is implied by $K \leq 2^{\mu}$. The right hand side of (4) being equal to $k/2^m$, (4) yields (3).

Case $j \leq t$. The proof is similar.

Case ($i \leq t$ and $t + l \leq j$). Let $\mathbf{b}_1, \ldots, \mathbf{b}_t$ be the words obtained from $\mathbf{c}_1, \ldots, \mathbf{c}_t$ by removing the first letter 0. These new words are all distinct and they constitute a maximal instantaneous code with this order of lexicographic enumeration and corresponding lexicographic length sequence $l_1 - 1, \ldots, l_t - 1$. Also, let $\mathbf{d}_{t+1}, \ldots, \mathbf{d}_n$ be the words obtained from $\mathbf{c}_{t+1}, \ldots, \mathbf{c}_n$ by removing the first letter 1, these are also distinct and constitute a maximal instantaneous code with this order of lexicographic enumeration and lexicographic length sequence $l_{t+1} - 1, \ldots, l_n - 1$. Further, $[i,t]$ is a full segment with respect to $l_1 - 1, \ldots, l_t - 1$, and [1, *j*-($t - i + 1$)] is full with respect to $l_{t+1} - 1, \ldots, l_n - 1$. Let μ_1 and μ_2 be the respective largest terms near these segments. Both μ_1 and μ_2 are at most $max\,(0, m-1)$. By the induction hypothesis there are integers $K_1 \leq 2^{\mu_1}$ and $K_2 \leq 2^{\mu_2}$ such that

$$\frac{1}{2^{l_i - 1}} + \cdots + \frac{l}{2^{l_t - 1}} = \frac{K_1}{2^{\mu_1}}$$

i.e.

$$\frac{1}{2^{l_i}}+\cdots+\frac{l}{2^{l_t}}=\frac{K_1}{2^{\mu_1+1}} \tag{5}$$

and

$$\frac{1}{2^{l_{t+1}-1}}+\cdots+\frac{l}{2^{l_j-1}}=\frac{K_2}{2^{\mu_2}}$$

i.e.

$$\frac{1}{2^{l_{t+1}}}+\cdots+\frac{l}{2^{l_j}}=\frac{K_2}{2^{\mu_2+1}} \tag{6}$$

If m=0 then $\mu_1=\mu_2=0$, $K_1=K_2=1$, the right hand sides of (5) and (6) are both 1/2, and adding (5) to (6) we get

$$\frac{1}{2^{l_i}}+\cdots+\frac{l}{2^{l_j}}=\frac{1}{2}+\frac{1}{2}=1=\frac{2^m}{2^m}$$

which gives (3) with k=1. If $m>0$ then both μ_1 and μ_2 are at most m-1. Let

$$k=\left(K_1+K_2 2^{\mu_1-\mu_2}\right)\cdot 2^{m-\mu_1-1} \text{ if } \mu_1\geq\mu_2$$

$$k=\left(K_1 2^{\mu_2-\mu_1}+K_2\right)\cdot 2^{m-\mu_2-1} \text{ if } \mu_1<\mu_2$$

It is easy to see, using $\mu_1+1\leq m$ and $\mu_2+1\leq m$, that $k\leq 2^m$. Also the sum of the right hand sides of (5) and (6) is $k/2^m$, thus adding (5) to (6) yields (3).

Conversely, for a sequence $l_1\ldots l_n$ satisfying equality (3) on its full segments, we show by induction on n that the sequence is the lexicographic length sequence of a maximal instantaneous code. This is clear for $n=1$, as (3) necessarily gives $l_1=0$, and the code is $\{\theta\}$. For $n>1$, and assuming the statement proved for lesser values, let $h=max(l_1\ldots l_n)$ and let i be the first index with $l_i=h$. Note that we must have $i<n$ and l_{i+1} also equal to h, for otherwise $[i,i]$ would be full and (3) would give, for some integer $m<h$

$$\frac{1}{2^h}=\frac{k}{2^m}$$

where $k<1$ would be an integer, which is impossible. By the Lemma it is sufficient to prove that the reduced sequence $l_1\ldots l_{i-1}hl_{i+2}\ldots l_n$ is a lexicographic length sequence. By the induction hypothesis it is enough to prove that this reduced sequence also satisfies the equality (3) an its full segments. Write

$$t_1=l_1,\ldots,t_{i-1}=l_{i-1},\ t_i=h,\ t_{i+1}=l_{i+2},\ldots,t_{n-1}=l_n$$

and let $[s,q]$ be a full segment will respect to $t_1, \ldots, t_{n-1}$. We distinguish the three cases $q < i$, $i < s$, and $s \leq i \leq q$.

Case $q < i$. It is not difficult to see that now $q < i - 1$ and $[s,q]$ is also a full segment with respect for the original sequence $l_1 \ldots l_n$, with the same largest term near it as in the reduced sequence $t_1 \ldots t_{n-1}$, and therefore satisfying (3) by the assumption or $l_1 \ldots l_n$.

Case $i < s$. The argument is similar.

Case $s < i < q$. By the definition of $h = l_i = l_{i+1}$, the segment $[s,q+1]$ is also full in $l_1 \ldots l_n$ and we have

$$\frac{1}{2^{l_s}} + \cdots + \frac{1}{2^{l_i}} + \frac{l}{2^{l_{i+1}}} + \cdots + \frac{l}{2^{l_{q+1}}} = \frac{k}{2^{\mu}} \tag{7}$$

when μ is the largest term near $[s,q+1]$ in $l_1 \ldots l_n$ and $k < 2^{\mu}$, k integer. It is easy to see that μ is also the largest term near $[s,q]$ in $t_1 \ldots t_{n-1}$. Equality (7) can be re-written as

$$\frac{1}{2^{l_s}} + \cdots + \frac{1}{2^{l_{i-1}}} + \frac{l}{2^h} + \frac{l}{2^h} + \frac{l}{2^{l_{i+2}}} + \cdots + \frac{l}{2^{l_{q+1}}} = \frac{k}{2^{\mu}}$$

and then

$$\frac{1}{2^{l_s}} + \cdots + \frac{1}{2^{l_{i-1}}} + \frac{l}{2^{h-1}} + \frac{l}{2^{l_{i+2}}} + \cdots + \frac{l}{2^{l_{q+1}}} = \frac{k}{2^{\mu}}$$

which is nothing else but (3) for the full segment $[s,q]$ of $t_1 \ldots t_{n-1}$.

□

References

[SPPR] D. Stott Parker, Prasad Ram, *The construction of Huffman codes is a submodular ("convex") optimization problem. SIAM J. Comput.* 28/5 (1999) 1875-1905

[R] Steven Roman, *Coding and Information Theory*, Springer 1992.