Generalizing the Kraft-McMillan Inequality to Restricted Languages (Extended Abstract)

Mordecai J. Golin* Hyeon-Suk Na†

Abstract

Let $\ell_1, \ell_2, \ldots, \ell_n$ be a (possibly infinite) sequence of nonnegative integers and Σ some D-ary alphabet. The *Kraft-Inequality* states that $\ell_1, \ell_2, \ldots, \ell_n$ are the lengths of the words in some prefix (free) code over Σ if and only if

$$\sum_{i=1}^{n} D^{-\ell_i} \leq 1. \tag{1}$$

Furthermore, the code is *exhaustive* if and only if equality holds. The *McMillan Inequality* states that if $\ell_1, \ell_2, \ldots, \ell_n$ are the lengths of the words in some uniquely decipherable code, then the same condition holds.

In this note we examine how the Kraft-McMillan Inequality conditions for the existence of a prefix or uniquely decipherable code change when the code is not only required to be prefix but all of the codewords are restricted to belong to a given specific language $\mathcal{L}$. For example, $\mathcal{L}$ might be all words that end in a particular pattern or, if Σ is binary, might be all words in which the number of zeros equals the number of ones.

1 Introduction

1.1 The Kraft-McMillan Inequalities

Let Σ be an alphabet. The *length* of word $w = \sigma_1\sigma_2 \ldots \sigma_\ell \in \Sigma^*$ is $|w| = \ell$. A word $w = \sigma_1\sigma_2 \ldots \sigma_\ell \in \Sigma^*$ is a *prefix* of another word $w' = \sigma'_1\sigma'_2 \ldots \sigma'_{\ell'} \in \Sigma^*$ if

$$\ell \leq \ell' \quad \text{and} \quad \forall i \leq \ell,\ \sigma_i = \sigma'_i.$$

A *code* C is a subset $C \subseteq \Sigma^*$. C is *prefix-free* if no word in C is a prefix of any other word in C, i.e.,

$$\forall w, w' \in C,\ w \text{ is not a prefix of } w'.$$

*Department of Computer Science, HKUST, Clear Water Bay, Kowloon, Hong Kong. Email: golin@cs.ust.hk

†School of Computing, Soongsil University, Seoul, South Korea. Email: hsnaa@computing.ssu.ac.kr

$C = \{000, 001, 01, 10, 11\}$
$E = (E_0, E_1, E_2, E_3) = (0, 0, 3, 2)$
$0 \cdot 2^0 + 0 \cdot 2^{-1} + 3 \cdot 2^{-2} + 2 \cdot 2^{-3} = 1$
An Exhaustive Code

$C' = \{000, 001, 01, 10\}$
$E' = (E'_0, E'_1, E'_2, E'_3) = (0, 0, 2, 2)$
$0 \cdot 2^0 + 0 \cdot 2^{-1} + 2 \cdot 2^{-2} + 2 \cdot 2^{-3} = \frac{3}{4} < 1$
Non-Exhaustive Code, since $C' \subsetneq C$

Figure 1: Two codes in $\{0,1\}^*$ and their corresponding trees. Note that left edges correspond to **0**s and right edges to **1**s. The depth of a leaf is equal to the number of characters in its corresponding node, e.g., the leftmost leaf has depth 3 and corresponds to word 000 which has length 3. Note too that the left code/tree is exhaustive while the right one isn't.

Such a code is sometimes also called a *prefix code.* For example {**11**, **10**, **01**, **001**, **000**} is a prefix (free) code while {**11**, **10**, **01**, **010**, **000**} is not.

Prefix-free codes are interesting for many reasons, e.g., they are uniquely decipherable and they often model searching or group-testing procedures.

There is a standard one-one correspondence between leaves of code-trees and prefix codes and, for the purposes of this paper, we will often be working with *trees* instead of codes. More specifically the correspondence is:

$$\begin{array}{rcl} \text{Edges leaving a node} & \leftrightarrow & \text{Characters in } \Sigma \\ \text{Leaves in tree} & \leftrightarrow & \text{Codewords} \\ \text{Node Depth} & \leftrightarrow & \text{Word Length} \end{array}$$

See for example, Figure 1. For convenience, in this note, we will often state our theorems in terms of code-trees rather than codes.

Definition 1 *For code $C \subseteq \Sigma^*$, define its* Level Sequence (L.S.) *to be $E = (E_0, E_1, \ldots)$ where $E_i = |\{w \in C : |w| = i\}|$. Note that E may be finite or infinite.*

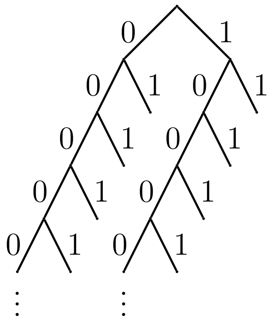
E is exhaustive *if there exists a prefix code with L.S. E but, $\forall j$, there does not exist a prefix code with L.S.*

$$E'_i = \begin{cases} E_i & \text{if } i \neq j \\ E_i + 1 & \text{if } i = j \end{cases}$$

The main result in this area (restated in terms of our notation) is

Theorem 1 (Kraft Inequality [10]) *Let $D = |\Sigma|$. There exists a prefix code C with level sequence E if and only if*

$$\sum_{i=0}^{\infty} E_i D^{-i} \leq 1. \tag{2}$$



$$\begin{aligned} C &= 0^*01 + 10^*1 \\ E &= (E_0, E_1, E_2, E_3, E_4 \ldots) \\ &= (0, 0, 2, 2, 2, \ldots) \end{aligned}$$

$$\sum_{i=0}^{\infty} E_i 2^{-i} = 2 \sum_{i=2}^{\infty} 2^{-i} = 1$$

Figure 2: An infinite tree corresponding to an exhaustive binary code.

E is exhaustive if and only if there is equality in (2), i.e., $\sum_{i=0}^{\infty} E_i D^{-i} = 1$.

The Kraft Inequality can also easily be read as a statement on how many leaves a D-ary tree can have on level ℓ (in which case the *Level Sequence* should be interpreted as a a *Leaf Sequence*). Besides prefix codes we can also discuss uniquely decipherable codes

Definition 2 *C is* Uniquely Decipherable (U.D.) *if every message in Σ^* can be decoded in at most one way using C.*

Every prefix code is obviously a U.D. code but not every U.D. code is prefix. For example, $C = \{0, 01, 11\}$ is not a prefix code but is U.D.. The major theorem concerning the existence of U.D. codes is

Theorem 2 (McMillan Inequality [12]) *If C is a U.D. code with level sequence E then*

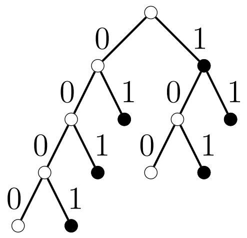
$$\sum_{i=0}^{\infty} E_i D^{-i} \leq 1.$$

Combining the Kraft and McMillan Inequalities we see that, if E is the level sequence of some U.D. code, then E must also be the level sequence of some prefix code (the converse is trivial since all prefix codes are U.D.).

1.2 Some Generalizations of Prefix Codes

The main result of this paper is to extend the Kraft Inequality to allow the code C to be restricted to be within some specified language $\mathcal{L} \subseteq \Sigma^*$. Before formally stating our problem and results we first illustrate with two examples from the literature.

The first example we discuss is *One-Ended Codes* due to Berger and Yeung [1] (see also [2, 3]). This is a code $C \subset \{0,1\}^*$ which, in addition to being prefix, also satisfies the condition that every codeword $w \in C$ *must end with a* **1**, i.e., $C \subseteq \mathcal{L} = (0+1)^*1$. Berger



i	0	1	2	3	4
E_i	0	0	2	2	1
$E_i 2^{-i}$	0	0	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{16}$

$$
\begin{array}{lll}
n=1 & E_1 2^0 + E_0 2^0 & = 0 \\
n=2 & E_2 2^{-1} + E_0 2^0 + E_1 2^{-1} & = 1 \\
n=3 & E_3 2^{-2} + E_0 2^0 + E_1 2^{-1} + E_2 2^{-2} & = 1 \\
n=4 & E_4 2^{-3} + E_0 2^0 + E_1 2^{-1} + E_2 2^{-2} + E_3 2^{-3} & = \frac{7}{8}
\end{array}
$$

Figure 3: An One-Ended Code whose level sequence E has five elements and how the E_i satisfy Eq (3). Note that this E is *not* exhaustive since, e.g., words **00001** and **1001** could be added to the code and it would remain prefix free. Black (filled) nodes denote *good* (one-ended) nodes. Note that there are 6 good nodes but only 5 good leaves (since node **1** is internal).

---

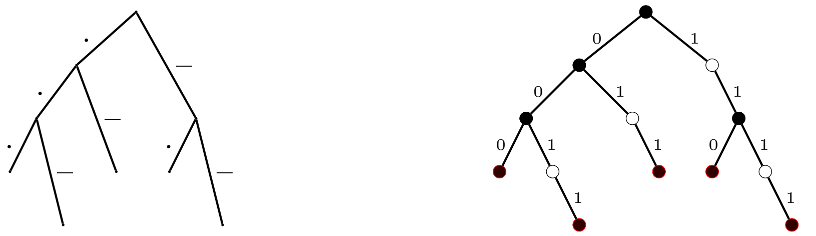
and Yeung proved that, E is a level sequence for One-Ended Codes if and only if E satisfies the following set of inequalities

$$\forall n, \quad E_n 2^{-(n-1)} + \sum_{i=0}^{n-1} E_i 2^{-i} \leq 1. \tag{3}$$

If $E = (E_0, E_1, \ldots, E_j)$ is finite, i.e., $\forall n > j$, $E_n = 0$, then Eq (3) only needs to be satisfied for $n \leq j$; if E is infinite, then Eq (3) is an infinite set of inequalities. Figure 3 gives an example of a finite One-Ended Code; we note that the infinite code in Figure 2 is also One-Ended and also satisfies (3).

As another example we discuss the unequal-cost coding problem, originally due to Karp [9] and sometime also known as Varn-Coding [16] (see [15, 4] for more historical references). In the standard prefix coding problem every character in Σ is considered to have the same length and the length of a word $w = \sigma_1 \sigma_2 \ldots \sigma_j \in \Sigma^*$ is simply the number of characters in w. In the unequal-cost coding problem each character in Σ can have a *different* length, and $|w| = \sum_{i=1}^{j} length(\sigma_i)$. This corresponds, for example, to cases in which characters have different transmission times or storage requirements; it also corresponds to searching problems in which different test-outcomes have different costs, e.g., [8].

One oft-quoted case is $\Sigma = \{\cdot, -\}$ where $length(\cdot) = \alpha$ and $length(-) = \beta$. When $(\alpha, \beta) = (1, 2)$, this is known as the *telegraph channel* [6]. We can use the same tree representation as before to represent prefix codes with the change that the *lengths of the edges* are no longer all identical but, instead, are equal to the length of the letter they represent. Using this correspondence the length of a word is the sum of lengths of its letters and the depth of a leaf is (still) the length of the path from the root to the leaf. With these definitions we can still continue to define the *level sequence* of code $C \subseteq \Sigma^*$ to be $E = (E_0, E_1, \ldots)$ where $E_i = |\{w \in C : |w| = i\}|$. E may be finite or infinite.



$C = \{\cdot\cdot\cdot, \cdot\cdot-, \cdot-, -\cdot, --\}$ $\qquad$ $C = \{000,\ 0011,\ 011,\ 110,\ 1111\}$

Figure 4: An unequal-cost tree in the telegraph channel and the corresponding binary tree along with their associated codes. In the binary tree the filled nodes are the good nodes (those corresponding to words in $\mathcal{L}$) while the unfilled nodes are not good. Note that both sets of trees/codes have the same level sequence $E = (E_0, E_1, E_2, E_3, E_4) = (0, 0, 0, 3, 2)$.

In [9] Karp looked at what was essentially the problem of generalizing the Kraft Inequality for unequal-cost letters under the assumption that the letter costs are all integral. He used generating function techniques to show that there exists a prefix code with finite level sequence E if and only if E satisfies a generating function identity (very similar to the sequence of convolutional inequalities we saw in Eq (3) for One-Ended Codes). He further showed that such a level sequence is exhaustive if and only if $\sum_i E_i \beta^{-i} = 1$ for some particular fixed β (dependent only upon the letter costs).

For the purposes of this note we point out that the unequal-cost coding problem (with integer letter costs) can be transformed to a language problem, see e.g., [7]. We illustrate this with the binary case. Suppose $\Sigma = \{\cdot, -\}$ where $length(\cdot) = \alpha$ and $length(-) = \beta$. This can be modelled with language $\mathcal{L} = \left(0^\alpha + 1^\beta\right)^*$; that is, every occurrence of $\cdot$ is replaced by α **0**s and every occurrence of $-$ is replaced by β **1**s. It is then easy to see that there exists an unequal-cost prefix code with level sequence E if and only if there exists a prefix code $C \subseteq \mathcal{L}$ with level sequence E. See Figure 4.

The purpose of this note is to develop a general theory for extending the Kraft Inequality in the direction that was illustrated above for the One-Ended and unequal-cost coding cases. We also show that there are many languages $\mathcal{L}$ for which it is provably not possible to develop an appropriate generalization of the Kraft Inequality. We conclude by giving a generalization of the McMillan Inequality (that is applicable in a smaller number of cases).

In the next section we first present our new results and then illustrate their application with various examples. *Due to space considerations in this extended abstract, proofs of the theorems have been omitted.*

2 Results and Examples

In this section we state our main results and then illustrate them with examples. Before doing so we will need to introduce some definitions and conditions.

We will use common notation, such as those found in [11], for operations on strings. In particular, if $\mathcal{A}, \mathcal{B} \subseteq \Sigma^*$ then their *concatenation* is $\mathcal{AB} = \{\alpha \circ \beta \,:\, \alpha \in \mathcal{A}, \beta \in \mathcal{B}\}$. $\mathcal{A}^i$ denotes the i-fold concatenation $\underbrace{\mathcal{AA}\ldots\mathcal{A}}_{i \text{ times}}$ with $\mathcal{A}^0$ being the empty word and the *Kleene star* is $\mathcal{A}^* = \bigcup_{i=0}^{\infty} \mathcal{A}^i$. Also, $\mathbf{0}/\mathbf{1}$ will denote the sets $\{0\}/\{1\}$ and $+$, set union. So, for example, $\mathbf{00}(\mathbf{1}+\mathbf{0})^*\mathbf{11}$ denotes all binary strings that start with $\mathbf{00}$ and end with $\mathbf{11}$.

2.1 Conditions

Definition 3 *Language $\mathcal{L} \subset \Sigma^*$ is* prefix *if $\forall w, w' \in \mathcal{L}$, w is not a prefix of w'. Language $\mathcal{L} \subset \Sigma^*$ is* memoryless *if*

$$\mathcal{L} = \mathcal{AB}^*$$

where $\mathcal{A}, \mathcal{B}$ are prefix.

Note that if $\mathcal{L}$ is memoryless, even though it has been built out of prefix languages, it usually isn't prefix itself. All of our later results will be on memoryless languages. This class is actually very powerful and contains all of the examples discussed before. More examples of memoryless languages are

- Let $P \in \Sigma^*$ be *any* string. Then the language $\mathcal{L} = \Sigma^* P$, the set of all words that end in P, is memoryless. In particular, if $\Sigma = \{0, 1\}$ and $P = 1$, then $\mathcal{L}$ is One-Ended words.
- Varn Languages (unequal-cost coding), e.g., $\mathcal{L} = \left(0^\alpha + 1^\beta\right)^*$.
- *Any* Language $\mathcal{L}$ accepted by a Deterministic Finite Automaton which has only one accepting state. The two previous examples are special cases of this case. See, e.g., [7]. This is memoryless by setting $\mathcal{A} = \{w \in \mathcal{L} \,:\, \text{no prefix } w' \text{ of } w \text{ is in } \mathcal{L}\}$, and $\mathcal{B} = \{v \in \Sigma^* \,:\, \forall w \in \mathcal{L},\, wv \in \mathcal{L} \text{ but } wv' \notin \mathcal{L} \text{ for any prefix } v' \text{ of } v\}$.
- Balanced Strings. $\mathcal{L} = \{w \in \{0,1\}^* \,:\, \text{No. of } \mathbf{0}\text{s in } w \text{ equals No. of } \mathbf{1}\text{s in } w\}$.
  In this case $\mathcal{A} = \{\epsilon\}$ and $\mathcal{B} = \{w \in \mathcal{L} \,:\, \text{no prefix } w' \text{ of } w \text{ is in } \mathcal{L}\}$.
  Note that, unlike the previous examples, this $\mathcal{L}$ is *not* a regular language.

We point out that if $\mathcal{L}$ is prefix, then by definition $\mathcal{B} = \{\epsilon\}$ and $\mathcal{L} = \mathcal{A}$. We will ignore this case in the sequel because restricting a prefix free code to be in a prefix language is not an interesting case.

Given a language $\mathcal{X}$ we will use the convention that $X_n = |\{w \in \mathcal{X} \,:\, |w| = n\}|$ and $X(z) = \sum_{n=0}^{\infty} X_n z^n$ will be the generating function of X_n. If $\mathcal{L} = \mathcal{AB}^*$ is memoryless then we will define the language $T = \mathcal{B}^*$ and introduce the four generating functions:

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{T} = \mathcal{B}^*$	$\mathcal{L} = \mathcal{AB}^*$
$A(z)$	$B(z)$	$T(z) = \frac{1}{1-B(z)}$	$L(z) = \frac{A(z)}{1-B(z)}$

(4)

Note that T_n will be the number of words of length n in $\mathcal{B}^*$ while L_n will be the number of length n in $\mathcal{L}$. The equations $T(z) = \frac{1}{1-B(z)}$ and $L(z) = \frac{A(z)}{1-B(z)}$ follow from the fact that $\mathcal{T} = \mathcal{B}^*$ and $\mathcal{L} = \mathcal{AT}$, and standard operations on generating functions, see, e.g., [5].

For later use we will need the following two properties:

Definition 4 (Unity Property) *Let $\mathcal{L} = \mathcal{A}\mathcal{B}^*$ be memoryless with $\mathcal{B} \neq \{\epsilon\}$. $\mathcal{L}$ satisfies the* Unity Property *if there exists real positive β such that $B(\beta) = 1$.*

Definition 5 (Algebraic Property) *Let $\mathcal{L} = \mathcal{A}\mathcal{B}^*$ be a memoryless language satisfying the* Unity Property *with $\beta > 0$, i.e., $B(\beta) = 1$. Set $\alpha = A(\beta)$. Then $\mathcal{L}$ satisfies the* Algebraic Property *if $\alpha < \infty$ and*

$$\forall n \geq 0, \quad T_n = f(n)\beta^{-n}\left[1 + o(1)\right] \quad \textit{and} \quad L_n = \alpha f(n)\beta^{-n}\left[1 + o(1)\right] \tag{5}$$

where $f(n)$ is a non-negative function that grows "polynomially", i.e., $\exists k \geq 0$ such that $f(n) = O(n^k)$.

It is beyond the scope of this extended abstract to describe the class of all languages that satisfy the two properties above. We restrict ourselves to pointing out that many interesting normal classes of languages satisfy these properties, e.g., any prefix-free $\mathcal{B}$ that is regular (i.e., accepted by a deterministic finite automaton) satisfies the unity property. If prefix-free $\mathcal{A}$ is such that $\alpha < \infty$ and $\mathcal{A}$ is also regular, then the algebraic property also holds. So, the first three examples given at the beginning of this section all satisfy the unity property and, if $\alpha < \infty$, also satisfy the algebraic property. It is not difficult to see that the balanced string example given previously also satisfies the two properties.

2.2 The Results

We can now state our main results.

Theorem 3 *Let $\mathcal{L} = \mathcal{A}\mathcal{B}^*$ be a memoryless language. There exists a prefix code $C \subseteq \mathcal{L}$ with level sequence $E = (E_0, E_1, \ldots)$ if and only if E satisfies the* convolutional identities (CI)

$$\forall n, \quad \sum_{i=0}^{n} E_i \, T_{n-i} \leq L_n. \tag{6}$$

Furthermore, if $\mathcal{L}$ satisfies the unity condition, let β be such that $B(\beta) = 1$ and set $\alpha = A(\beta)$. If E satisfies the CI, then

$$\sum_{i=0}^{\infty} E_i \beta^i \leq \alpha. \tag{7}$$

One can think of this theorem as splitting the original Kraft Inequality up into two parts. The first, (6), gives a necessary and sufficient condition for the existence of a prefix code. The second, (7), gives a numerical inequality that is a consequence.

Definition 6 *Let $\mathcal{L}$ be any language. We say that $\mathcal{L}$ satisfies the $(\hat{\alpha}, \hat{\beta})$-*Kraft Condition *if, for every E that is the L.S. of some prefix code $C \subseteq \mathcal{L}$,*

E is Exhaustive if and only if $\sum_{i=0}^{\infty} E_i \hat{\beta}^i = \hat{\alpha}$.

Theorem 4 *Let $\mathcal{L} = \mathcal{A}\mathcal{B}^*$ be a memoryless language satisfying the unity and algebraic properties with $B(\beta) = 1$ and $\alpha = A(\beta)$. Furthermore, suppose $\alpha < \infty$. Then, $\mathcal{L}$ satisfies the (α, β)-Kraft Condition.*

Theorem 5 *Let $\mathcal{L} = \mathcal{A}\mathcal{B}^*$ be a memoryless language satisfying the unity property with $B(\beta) = 1$. Set $\alpha = A(\beta)$. If $\alpha = \infty$ then there does not exist* any *$(\hat{\alpha}, \hat{\beta})$ pair such that $\mathcal{L}$ satisfies the $(\hat{\alpha}, \hat{\beta})$-Kraft Condition.*

Theorem 6 *Let $\mathcal{L} = \mathcal{A}\mathcal{B}^*$ be a memoryless language satisfying the unity and algebraic properties with $B(\beta) = 1$ and $\alpha = A(\beta) = 1$. Then*

If E is the L.S. of a uniquely decipherable code then $\sum_{i=0}^{\infty} E_i \beta^i \leq 1$

We point out that even though the condition $\alpha = 1$ is quite restrictive it still encompasses many languages. For example, if $\mathcal{A} = \{\epsilon\}$, then $A(z) = 1$ and the condition $\alpha = A(\beta) = 1$ is trivially satisfied, so the unequal-cost coding problems discussed before all have $\alpha = 1$.

2.3 Examples

In this section we give some examples of the application of our results. We start with a "reality-check" by showing that, in the unrestricted case $\mathcal{L} = \Sigma^*$, our theorems collapse down to the standard Kraft-McMillan Inequalities given in Theorems 1 and 2.

Example: The standard Kraft-McMillan Inequalities when $|\Sigma| = D$.
In this case $\Sigma^* = \mathcal{L}$ so $\mathcal{A} = \{\epsilon\}, \mathcal{B} = \Sigma$, $\mathcal{L} = \mathcal{T} = \mathcal{B}^*$ and $L_n = T_n = D^n$. Working through the formulas gives $A(z) = 1$, $B(z) = Dz$, $\beta = 1/D$, $\alpha = A(1/D) = 1$ and $\mathcal{L}$ satisfies the unity and algebraic properties. Theorem 3 then says that E is prefix-free iff

$$\forall n,\ \sum_{i=0}^{n} E_i\, T_{n-i} \leq L_n \quad \Longleftrightarrow \quad \forall n,\ \sum_{i=0}^{n} E_i\, D^{n-i} \leq D^n \quad \Longleftrightarrow \quad \forall n,\ \sum_{i=0}^{n} E_i\, D^{-i} \leq 1$$

which collapses down to the single equation $\sum_{i=0}^{\infty} E_i\, D^{-i} \leq 1$. Theorem 4 says that if E satisfies this condition then E is exhaustive if and only if $\sum_{i=0}^{n} E_i\, D^{-i} = 1$. Combining the two just gives the standard Kraft Inequality from Theorem 1. Finally, Theorem 6 says that if E is uniquely decipherable then $\sum_{i=0}^{n} E_i\, D^{-i} \leq 1$ which is just the standard McMillan result of Theorem 2.

Example: One-Ended Codes. $\mathcal{L} = (0+1)^*1$.
In this case $\mathcal{A} = \mathcal{B} = 0^*1$ and $\mathcal{T} = \mathcal{B}^* = \epsilon + (0+1)^*1$. Then

$$A_n = B_n = \left\{ \begin{array}{ll} 0 & \text{if } n = 0. \\ 1 & \text{if } n > 0. \end{array} \right\}, \quad T_n = \left\{ \begin{array}{ll} 1 & \text{if } n = 0. \\ 2^{n-1} & \text{if } n > 0. \end{array} \right\}, \quad L_n = \left\{ \begin{array}{ll} 0 & \text{if } n = 0. \\ 2^{n-1} & \text{if } n > 0. \end{array} \right\}$$

Thus $A(z) = B(z) = \sum_{n=1}^{\infty} z^n = \frac{z}{1-z}$, $\beta = 1/2$ and $A(1/2) = B(1/2) = 1$. Theorem 3 says that E is prefix-free iff

$$\forall n \quad \sum_{i=0}^{n} E_i T_{n-i} \leq L_n \Longleftrightarrow E_n + \sum_{i=0}^{n-1} E_i 2^{(n-i)-1} \leq 2^{n-1} \Longleftrightarrow E_n 2^{-(n-1)} + \sum_{i=0}^{n-1} E_i 2^{-i} \leq 1$$

which is *exactly* the result of Berger and Yeung [1] that we saw in equation (3). Since $\mathcal{L}$ satisfies both the unity and algebraic properties, Theorem 4 tell us, in addition, that E satisfying the CI is Exhaustive iff $\sum_{i=0}^{\infty} E_i 2^{-i} = 1$. Furthermore, since $\alpha = 1$, Theorem 6 tells us that if a One-Ended Code is uniquely decipherable then its level sequence E must satisfy $\sum_{i=0}^{\infty} E_i 2^{-i} \leq 1$.

Example: $(1, 2)$-unequal cost codes. $\mathcal{L} = (0 + 11)^*$.
In this case $\mathcal{B} = 0+11$, $\mathcal{A} = \{\epsilon\}$ and $\mathcal{L} = \mathcal{T} = \mathcal{B}^*$. It is then easy to derive that $L_n = T_n = F_n$ where F_n are the Fibonacci numbers. $B(z) = z + z^2$, $\beta = \frac{\sqrt{5}-1}{2} = \frac{1}{\phi}$, where $\phi = \frac{\sqrt{5}+1}{2}$. Since $A(z) = 1$, $\alpha = A(\beta) = 1$. Theorem 3 says that E is prefix-free iff

$$\forall n, \quad \sum_{i=1}^{n} E_i F_{n-i} \leq F_n \tag{8}$$

and that if E does satisfy (8), then $\sum_{i=0}^{\infty} E_i\, \phi^{-i} \leq 1$. Since $\mathcal{L}$ satisfies both the unity and algebraic properties, Theorem 4 tell us, in addition, that E satisfying (8) is Exhaustive iff $\sum_{i=0}^{\infty} E_i\, \phi^{-i} = 1$. This was all actually implicit in Karp's analysis in [9]. Finally, Theorem 6 permits us to derive the new result that if E is the L.S. of a uniquely decipherable $(1, 2)$-unequal cost code then $\sum_{i=0}^{\infty} E_i\, \phi^{-i} \leq 1$.

We conclude with two arbitrary examples that illustrate the case $\alpha \neq 1$ and the case that $\mathcal{L}$ does not satisfy the algebraic property. For these cases we do not explicitly state the values T_n and L_n.

Example: $\mathcal{L} = (0 + 11 + 1001)^*2(0^*1)^*$.
In this case $\mathcal{A} = (0 + 11 + 1001)^*2$, and $\mathcal{B} = 0^*1$, so $B(z) = \frac{z}{1-z}$, $\beta = 1/2$, $A(z) = \frac{z}{1-z-z^2-z^4}$, and $A(\beta) = \frac{8}{3}$. If E satisfies the CI (6), then $\sum_{i=0}^{\infty} E_i 2^{-i} \leq \frac{8}{3}$ and E is exhaustive iff $\sum_{i=0}^{\infty} E_i 2^{-i} = \frac{8}{3}$

Example: $\mathcal{L} = (0 + 1)^*2(0 + 11)^*$.
In this case $\mathcal{A} = (0+1)^*2$, $\mathcal{B} = 0+11$. Then $B(z) = z+z^2$, $\beta = \frac{1}{\phi} \sim 0.618$, $A(z) = \frac{z}{1-2z}$. The important fact here is that β *is greater than the radius of convergence of* $A(z)$ so $A(\beta) = \infty$. Since $\alpha = \infty$, Theorem 5 tells us that there do not exist *any* $(\hat{\alpha}, \hat{\beta})$ pair such that if E satisfies the CI then "E is exhaustive for $\mathcal{L}$ iff $\sum_{i=0}^{\infty} E_i \hat{\beta}^i = \hat{\alpha}$".

3 Conclusion

In this note we presented some generalizations of the Kraft-McMillan inequalities for prefix-codes restricted to be within specified languages $\mathcal{L}$. The major open questions that remain are whether it is possible to extend our generalizations even further, e.g., by weakening the various conditions, such as the requirement of the unity and algebraic properties defined in subsection 2.1. In particular, the question of developing an efficient, e.g., polynomial time, procedure for identifying whether a given finite E is achievable as the level sequence for a prefix code in a *non-memoryless* language $\mathcal{L}$ seems to be fully open.

Acknowledgments: The authors would like to thank Dominique Perrin, Raymond Yeung and H.-K. Hwang for various comments and suggestions. They would also like to thank Kin Keung Ma for his assistance in drawing some of the figures. This paper was partially supported by HK RGC CERG grants 6162/00E, 6082/01E and RGC Direct Allocation Grant DAG03/04.EG06.

References

[1] T. Berger and R. W. Yeung, "Optimum "1"-ended Binary Prefix codes," *IEEE Transactions on Information Theory*, **36** (6) (Nov. 1990), 1435-1441.

[2] R. M. Capocelli, A. De Santis, and G. Persiano, "Binary Prefix Codes Ending in a "1"," *IEEE Transactions on Information Theory*, **40** (1) (Jul. 1994), 1296-1302.

[3] Chan Sze-Lok and M. Golin, "A Dynamic Programming Algorithm for Constructing Optimal "1"-ended Binary Prefix-Free Codes," *IEEE Transactions on Information Theory,* **46**(4) (July 2000) 1637-44.

[4] V. S.-N. Choi and Mordecai J. Golin, "Lopsided Trees I: Analyses," *Algorithmica,* **31**, (2001) 240-290.

[5] Philippe Flajolet and Robert Sedgewick, *An Introduction to the Analysis of Algorithms,* Addison-Wesley. (1995).

[6] E.N. Gilbert, "Coding with Digits of Unequal Cost," *IEEE Transactions on Information Theory,* **41**(2) (March 1995) 596-600.

[7] M. Golin and HyeonSuk Na, "Optimal prefix-free codes that end in a specified pattern and similar problems: the uniform probability case.," *IEEE Data Compression Conference, DCC'2001,* (March 2001) 143-152.

[8] Sanjiv Kapoor and Edward M. Reingold, "Optimum Lopsided Binary Trees," *Journal of the Association for Computing Machinery,* **36**(3) (July 1989) 573-590.

[9] R. M. Karp, "Minimum-redundancy coding for the discrete noiseless channel," *IRE Transactions on Information Theory,* **7**, (1961) 27-39.

[10] L.G. Kraft, *A Device for Quantizing, Grouping, and Coding Amplitude Modulated Pulses.* M.S. Thesis, Electrical Engineering Department, MIT (1949).

[11] Harry R. Lewis and Christos H. Papadimitriou, *Elements of the Theory of Computation (2nd ed.),* Prentice Hall. (1998).

[12] Brockway McMillan, "Two inequalities implied by unique decipherability." *IRE Transactions on Information Theory,* **2**, (Dec 1956) 115 - 116 .

[13] Y. Perl, M. R. Garey, and S. Even. "Efficient Generation of Optimal Prefix Code: Equiprobable Words Using Unequal Cost Letters," *Journal of the Association for Computing Machinery*, **22**(2) (April 1975) 202–214.

[14] Serap A. Savari, "Some Notes on Varn Coding," *IEEE Transactions on Information Theory,* **40**(1) (Jan. 1994) 181-186.

[15] Serap A. Savari, "A Probabilistic Approach to Some Asymptotics in Noiseless Communications," *IEEE Transactions on Information Theory,* **46**(4) (July 2000) 1246-1262.

[16] B.F. Varn, "Optimal Variable Length Codes (Arbitrary Symbol Costs and Equal Code Word Probabilities)," *Informat. Contr.,* **19** (1971) 289-301.