ISIT'2013 1569719125

Redundancy Analysis in Lossless Compression of a Binary Tree Via its Minimal DAG Representation

Jie Zhang
Avoca Technologies Inc.
563 Edward Ave., Suite 13
Richmond Hill, Ontario CA
Email: jie@avocatec.com

En-hui Yang
ECE Department
University of Waterloo
Waterloo, Ontario CA
Email: ehyang@uwaterloo.ca

John C. Kieffer
ECE Department
University of Minnesota Twin Cities
Minneapolis, MN
Email: kieffer@umn.edu

***Abstract*—Let $\mathcal{T}$ denote the set of all structurally inequivalent finite rooted ordered binary trees. For each $t \in \mathcal{T}$, let $D(t)$ be the unique minimal DAG representation of t, and let $r(t) \in (0,1]$ be the ratio of the number of vertices of $D(t)$ to the number of leaves of t. A lossless prefix encoder ϕ on $\{D(t) : t \in \mathcal{T}\}$ is proposed, and then a two-step lossless encoder ϕ^* on $\mathcal{T}$ is defined by $\phi^*(t) \triangleq \phi(D(t))$ for $t \in \mathcal{T}$. Let γ be the function $\gamma(x) \triangleq (x/2)\log_2(2/x)$ for $x \in (0,1]$. It is shown that the normalized pointwise redundancy in encoding each $t \in \mathcal{T}$ via ϕ^* is $O(\gamma(r(t)))$. Furthermore, given a binary tree source whose output is a sequence of random trees growing in size, weak sufficient conditions on the source are presented under which the normalized average redundancy of ϕ^* with respect to the source vanishes asymptotically. This result allows for the identification of some families of binary tree sources on which ϕ^* acts as a universal code.**

I. Introduction

There have been some initial efforts in the development of a lossless compression theory for structures [6][2]. In the present paper, we investigate the efficacy of a lossless compression scheme for binary tree structures proposed in [7][1]. By "binary tree" we mean throughout a rooted tree with finitely many vertices in which each non-leaf vertex has exactly two ordered vertices as its children; we regard each binary tree as a directed graph in which the direction along each edge is away from the root. When we say that two binary trees t_1, t_2 are isomorphic, we mean that there is a one-to-one map from the vertex set of t_1 onto the vertex set of t_2 which preserves the ordered tree structure. We shall only be interested in sets of binary trees which are structurally inequivalent, by which we mean that no two distinct trees in the set are isomorphic. It is well known [8] that for each $n \geq 1$ we may choose a structurally inequivalent set $\mathcal{T}_n$ consisting of $n^{-1}\binom{2(n-1)}{n-1}$ binary trees each having n leaves such that each binary tree having n leaves is isomorphic to a unique tree in $\mathcal{T}_n$. For $n \geq 1$, each tree in $\mathcal{T}_n$ has precisely $n-1$ non-leaf vertices and $2(n-1)$ edges; see, for example, Fig. 1 for the case $n = 4$. The set $\mathcal{T}_1$ consists of just one tree t^*, where t^* has only one vertex (which is both a leaf and the root) and no edges. We let $\mathcal{T} = \cup_{n=2}^{\infty}\mathcal{T}_n$ and let $\mathcal{T}^* = \{t^*\} \cup \mathcal{T}$.

Given $t \in \mathcal{T}$, a rooted DAG (directed acyclic graph) with two ordered outgoing edges from each non-leaf vertex whose root-to-leaf paths are in one-to-one correspondence with the root-to-leaf paths of t is called a DAG representation of t; this one-to-one correspondence of paths allows one to reconstruct a tree $t \in \mathcal{T}$ from any of its DAG representations. (For example, the tree on the left in Fig. 2 and its DAG representaton on the right have 8 corresponding pairs of root-to-leaf paths, each pair of paths having the same sequence of edge labels; one of these pairs has sequence of edge labels $1,1,2,2$, and so on for the other 7 corresponding pairs.) Each $t \in \mathcal{T}$ has a unique DAG representation (up to isomorphism) with the minimal number of vertices [7]; this DAG shall be referred to as the minimal DAG representation of t. Fig. 2 depicts a binary tree and its minimal DAG representation; as in Fig. 2, minimal DAG representations have only one leaf, since minimization of the number of vertices entails that all leaves be merged.

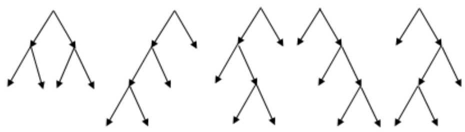


Fig. 1: The 5 trees in $\mathcal{T}_4$; each has 4 leaves, 3 non-leaves, and 6 edges.

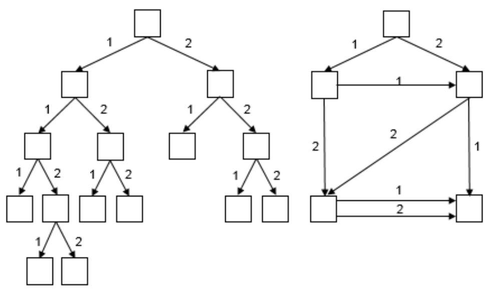


Fig. 2: A binary tree (left) and its minimal DAG representation (right)

There are linear-time algorithms for computing minimal DAG representations of binary trees; one of them is discussed in [1]. Although space does not permit us to discuss any of these algorithms, it is straightforward to specify the structure of the minimal DAG representation, which we do in the following discussion. For each $t \in \mathcal{T}$, if v is a vertex of t, define $t(v)$ to be the unique tree in $\mathcal{T}^*$ which is isomorphic to the subtree of t rooted at v whose vertices consist of v

together with the vertices of t which are descendants of v in t. (Note that $t(v) = t^*$ if v is a leaf of t.) Also, define t_L, t_R, respectively, to be the $t(v)$ trees in which v ranges through the ordered children of the root of t. Given $t \in \mathcal{T}$ with set of vertices $V(t)$, let D^* be the rooted directed acyclic graph such that

- The set of vertices of D^* is $\{t(v) : v \in V(t)\}$.
- The root vertex of D^* is t.
- The only leaf vertex of D^* is t^*.
- For each non-leaf vertex u of D^*, there are two ordered edges of D^* emanating from u, which respectively terminate at vertices u_L, u_R of D^*.

The minimal DAG representation of t can be taken to be any directed acyclic graph isomorphic to D^*; in Sec. II, we will choose a particular such graph, which we denote by $D(t)$, as *the* minimal DAG representation of t.

Representation Ratio. $|t|$ shall denote the number of leaves of $t \in \mathcal{T}^*$, and for each $t \in \mathcal{T}$, $|D(t)| \geq 2$ shall denote the number of vertices of $D(t)$. We define the *representation ratio* of $t \in \mathcal{T}$ to be the ratio

$$r(t) \triangleq \frac{|D(t)|}{|t|},$$

which is easily shown to belong to the interval $(0, 1]$. For example, for the tree t in Fig. 2, $r(t) = 5/8$. The representation ratio $r(t)$ has important implications for lossless compression of binary tree t. To see this, first note that, since t is recoverable from $D(t)$, one can losslessly compress t in two steps by (1) forming $D(t)$ from t in linear time, and (2) compressing $D(t)$ into a unique binary codeword and then using this codeword as the codeword for t; the compression scheme is lossless because in decoding step (1) one can decode $D(t)$ from its codeword and in decoding step (2) one can recover t from $D(t)$. For a binary tree t for which the representation ratio $r(t)$ is considerably small, it has been suggested [7] [1] that this two-step compression scheme will provide efficient compression of t. We make this intuition precise in our main result, Theorem 1, which states that the normalized pointwise redundancy of the two-step scheme in encoding each $t \in \mathcal{T}$ is upper bounded by a constant times a nonlinear function $\gamma(r(t))$ of $r(t)$ which tends to zero as $r(t)$ tends to zero. Before giving the precise form of Theorem 1, we need to make a series of definitions.

Definitions.

- A lossless prefix encoder on a countable set $\mathcal{S}$ is defined as a one-to-one mapping α from $\mathcal{S}$ into the set of finite-length binary strings such that if s_1, s_2 are any two distinct elements of $\mathcal{S}$, then $\alpha(s_1)$ is not a prefix of $\alpha(s_2)$.
- Let $\mathcal{D} = \{D(t) : t \in \mathcal{T}\}$. If ϕ is a lossless prefix encoder on $\mathcal{D}$, then we define ϕ^* to be the lossless prefix encoder on $\mathcal{T}$ such that

  $$\phi^*(t) = \phi(D(t)), \;\; t \in \mathcal{T}.$$

  ϕ^* is called a two-step encoder (induced by ϕ).
- If b is a binary string, let $L[b]$ denote the length of b.
- Let $\gamma : (0, 1] \to (0, \infty)$ be the function

  $$\gamma(x) \triangleq (x/2)\log_2(2/x), \quad x \in (0, 1].$$

- Let Λ be the set of all functions $\lambda : \mathcal{T}^* \to (0, 1]$ such that
  - **(a):** For each $t \in \mathcal{T}$,

    $$\lambda(t) \leq \lambda(t_L)\lambda(t_R). \tag{1}$$

  - **(b):** There exists a positive integer $K(\lambda)$ such that

    $$1 \leq \sum_{t \in \mathcal{T}_n} \lambda(t) \leq n^{K(\lambda)}, \;\; n \geq 1. \tag{2}$$

Theorem 1. There exists a two-step encoder ϕ^* on $\mathcal{T}$ such that for any $\lambda \in \Lambda$, there exists $C(\lambda) \in (0, \infty)$ depending only upon λ such that

$$|t|^{-1}\{L[\phi^*(t)] + \log_2 \lambda(t)\} \leq C(\lambda)\gamma(r(t)), \;\; t \in \mathcal{T}. \tag{3}$$

A two-step encoder on $\mathcal{T}$ for which Theorem 1 holds will be defined in Sec. III; Theorem 1 will be proved in Sec. IV. The left side of the bound (3) is referred to as the *normalized pointwise redundancy* in encoding individual tree t via ϕ^* (with respect to λ).

Application to Binary Tree Source Coding. We define a binary tree source as any pair (F, P) in which

- $F = (F_i : i \geq 1)$, a sequence of nonempty finite subsets of $\mathcal{T}$ which form a partition of $\mathcal{T}$.
- P is a mapping from $\mathcal{T}$ into $[0, 1]$ whose restriction to each F_i is a probability distribution on F_i.

Example 1. Let $F^{(1)} = (F_i^{(1)} : i \geq 1)$ be the sequence in which $F_i^{(1)} = \mathcal{T}_{i+1}$. Let σ be a mapping from $\{1, 2, \cdots\} \times \{1, 2, \cdots\}$ into $[0, 1]$ such that

$$\sum_{\{(i,j): i,j \geq 1,\, i+j=n\}} \sigma(i, j) = 1, \;\; n \geq 2.$$

Let P_σ be the mapping from $\mathcal{T}$ into $[0, 1]$ such that

$$P_\sigma(t) = \prod_{v \in V^1(t)} \sigma(|t(v)_L|, |t(v)_R|), \;\; t \in \mathcal{T},$$

where $V^1(t)$ is the set of non-leaf vertices of t. $(F^{(1)}, P_\sigma)$ is called a *leaf-centric* binary tree source [6][9].

Example 2. Let $d(t)$ be the maximal depth of $t \in \mathcal{T}^*$ (meaning that the longest root-to-leaf path in t consists of $d(t)$ edges). Let $F^{(2)} = (F_i^{(2)} : i \geq 1)$ be the sequence in which $F_i^{(2)}$ is the set of $t \in \mathcal{T}$ for which $d(t) = i$. Let σ be a mapping from $\{0, 1, \cdots\} \times \{0, 1, \cdots\}$ into $[0, 1]$ such that

$$\sum_{\{(i,j): i,j \geq 0,\, \max(i,j)=n-1\}} \sigma(i, j) = 1, \;\; n \geq 1.$$

Let P_σ be the mapping from $\mathcal{T}$ into $[0, 1]$ such that

$$P_\sigma(t) = \prod_{v \in V^1(t)} \sigma(d(t(v)_L), d(t(v)_R)), \;\; t \in \mathcal{T}.$$

$(F^{(2)}, P_\sigma)$ is called a *depth-centric* binary tree source [9].

Given a binary tree source (F, P) and a lossless prefix encoder ψ on $\mathcal{T}$, we define the i-th order normalized average redundancy of ψ with respect to the source to be the real number

$$R_i(\psi,F,P) \triangleq \sum_{t\in F_i,\,P(t)>0} |t|^{-1}\{L[\psi(t)]+\log_2 P(t)\}P(t).$$

We say that ψ is an asymptotically optimal encoder for the source (F,P) if

$$\limsup_{i\to\infty} R_i(\psi,F,P) \le 0. \tag{4}$$

Theorem 2. Let (F,P) be any binary tree source satisfying the following two conditions:

- **Condition 1:** There exists $\lambda \in \Lambda$ such that

$$P(t) \le \lambda(t), \;\; t \in \mathcal{T}. \tag{5}$$

- **Condition 2:** $\lim_{i\to\infty}\sum_{t\in F_i} r(t)P(t) = 0$.

Then the two-step encoder ϕ^* of Theorem 1 is asymptotically optimal for source (F,P).

Proof. Letting $\lambda \in \Lambda$ satisfy (5), we have from (3) and the concavity of γ that

$$R_i(\phi^*,F,P) \le C(\lambda)\gamma\left(\sum_{t\in F_i} r(t)P(t)\right), \;\; i \ge 1,$$

from which (4) follows for $\psi = \phi^*$ by exploiting Condition 2.

Application to Universal Coding. Let $F = (F_i : i \ge 1)$ be a fixed sequence of finite nonempty subsets of $\mathcal{T}$ forming a partition of $\mathcal{T}$. Let $\mathbb{P}$ be a family of mappings from $\mathcal{T}$ into $[0,1]$ such that each of them, restricted to each F_i, yields a probability distribution on F_i. A lossless prefix encoder ψ on $\mathcal{T}$ is defined to be a universal code for the family of binary tree sources $\{(F,P) : P \in \mathbb{P}\}$ if it is asymptotically optimal for each source in the family. Theorem 2 tells us that the lossless prefix encoder ϕ^* of Theorem 1 will be a universal code for family $\{(F,P) : P \in \mathbb{P}\}$ if the weak Conditions 1-2 hold for every source in the family. In this way, the paper [9] is able to isolate a large family of leaf-centric binary tree sources and a large family of depth-centric binary tree sources for which ϕ^* is a universal code. For example, [9] shows that ϕ^* is universal for the family of all leaf-centric sources $(F^{(1)},P_\sigma)$ for which

$$\sup\left\{\frac{i+j}{\min(i,j)} : i,j \ge 1,\;\sigma(i,j) > 0\right\} < \infty.$$

II. DAG Representation $D(t)$ of Tree t

Let $t \in \mathcal{T}$. We now define the particular directed acyclic graph $D(t)$ that we employ as the minimal DAG representation of t. In Sec. I, we defined a directed acyclic graph D^* whose set of vertices is $\{t(v) : v \in V(t)\}$ which serves as a minimal DAG representation of t. The graph $D(t)$ can then simply be defined by re-labeling the vertices of D^*. Let K be the cardinality of the set of subtrees $\{t(v) : v \in V^1(t)\}$. The non-leaf vertices of $D(t)$ are $0,1,\cdots,K-1$, with 0 the root vertex, and the single leaf vertex of $D(t)$ is a special symbol T. There is a one-to-one correspondence between the sets $\{t(v) : v \in V^1(t)\}$ and $\{0,1,\cdots,K-1\}$ obtained as follows. First, let $v_0,v_1,\cdots,v_j$ be the list of the non-leaf vertices of t in breadth-first order. Then, form the list

$$t(v_0),t(v_1),\cdots,t(v_j) \tag{6}$$

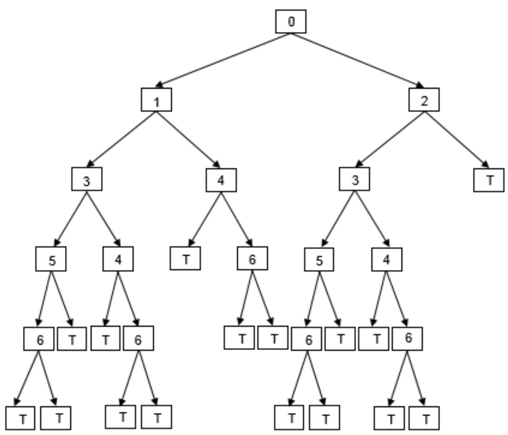
and traverse this list from left-to-right. The first entry $t(v_0)$ is t, which we label as t_0. We let $t_1 = t(v_1)$. We define t_2 to be the first member encountered left-to-right in the list (6) which does not belong to $\{t_0,t_1\}$. Continuing in this way, suppose $t_0,t_1,\cdots,t_m$ have been defined, where $m < K-1$. Then t_{m+1} is defined to be the first entry encountered left-to-right in the list (6) which does not belong to the set $\{t_1,t_2,\cdots,t_m\}$. When the labeling has been finished, the set $\{t(v) : v \in V^1(t)\}$ takes the form $\{t_0,t_1,\cdots,t_{K-1}\}$, and we thus have the desired correspondence $t_i \leftrightarrow i$. In the directed acyclic graph D^*, re-label each non-leaf t_i as i and re-label the leaf t^* as the special symbol T; this gives us an isomorphism carrying the graph D^* onto the graph $D(t)$.

For each non-leaf vertex $i \in \{0,1,\cdots,K-1\}$ of $D(t)$, let $\alpha(i,1),\alpha(i,2)$ be the respective vertices of $D(t)$ at the terminus of the two ordered edges of $D(t)$ which emanate from i. To specify $D(t)$, it suffices to form the list of rules

$$i \to (\alpha(i,1),\alpha(i,2)), \;\; i = 0,1,\cdots,K-1. \tag{7}$$

Here is an easy way to obtain the list of rules (7). Traverse the vertices of t in breadth-first order. Along the way, label each vertex v with the element of $\{0,1,\cdots,K-1\}$ corresponding to $t(v)$ if $v \in V^1(t)$; otherwise, label v by T. To obtain the rule in list (7) whose left side is i, simply find any vertex v of t whose label is i; the two ordered children v_1,v_2 of v then have labels $\alpha(i,1),\alpha(i,2)$, respectively.

Example 3. Let t be the binary tree in Fig. 3 (without the vertex labels). The vertex labels assigned in Fig. 3 were done according to the methodology of the preceding paragraph. From the labeled tree, it is easy to pick off the list of rules of $D(t)$, which appear on the right in Fig. 3.



$0 \to (1,2)$

$1 \to (3,4)$

$2 \to (3,T)$

$3 \to (5,4)$

$4 \to (T,6)$

$5 \to (6,T)$

$6 \to (T,T)$

Fig. 3: Labeling of Ex. 3 tree t (left) to form rules of $D(t)$ (right)

III. Encoding DAG Representations

Fix $D \in \mathcal{D} = \{D(t) : t \in \mathcal{T}\}$. We now specify a unique binary codeword $\phi(D)$ for D, via a technique which is an adaptation of the grammar encoding technique employed in [5]. The resulting encoder ϕ on $\mathcal{D}$ will then be a lossless prefix encoder, and the two-step encoder ϕ^* on $\mathcal{T}$ induced by ϕ can be taken as the two-step encoder in Theorem 1.

The set of non-leaf vertices of D is $\{0,1,\cdots,|D|-2\}$, and the single leaf vertex of D is the special symbol T. We let $V^*(D)$ denote the set of non-leaf vertices of D other than the root; we have $V^*(D)=\{1,2,\cdots,|D|-2\}$. Let (7) be the list of $|D|-1$ rules of D which specifies D as described in Sec. II. Let $S(D)$ be the sequence of length $2|D|-2$ whose first two entries are $\alpha(0,1),\alpha(0,2)$, whose next two entries are $\alpha(1,1),\alpha(1,2)$, and so forth. The alphabet of the sequence $S(D)$ is $V^*(D)\cup\{T\}$, which is of cardinality $|D|-1$. For each $v\in V^*(D)\cup\{T\}$, let $f_v>0$ denote the number of entries of $S(D)$ equal to v. $S(D)'$ is defined as the sequence obtained from $S(D)$ by striking out each of the $|D|-2$ entries of $S(D)$ which belongs to $V^*(D)$ and is making its first left-to-right appearance in $S(D)$. $S(D)'$ therefore has length $2(|D|-1)-(|D|-2)=|D|$. Let $M(D)$ denote the integer

$$M(D)\triangleq\frac{|D|!}{f_T!\prod_{v\in V^*(D)}(f_v-1)!},$$

and let $\mathcal{S}(D)'$ be the set of $M(D)$ sequences of length $|D|$ in which the frequency of T is f_T and the frequency of each $v\in V^*(D)$ is f_v-1; the sequence $S(D)'$ belongs to $\mathcal{S}(D)'$ and $\mathcal{S}(D)'$ consists of all permutations of $S(D)'$.

The idea of forming the binary codeword $\phi(D)$ is to start it with some bits denoting what $|D|$ is, followed by some bits indicating what $S(D)$ is. In decoding, one determines $|D|$ and $S(D)$ from codeword $\phi(D)$, from which the list of rules (7) is then formed, determining D. If $|D|=2$ (which means that the tree represented by D is the unique binary tree in $\mathcal{T}$ having two leaves), we define the codeword $\phi(D)$ to be 1. Assume from now on that $|D|>2$. Then we define $\phi(D)$ to be the left-to-right concatenation of the four binary strings B_1,B_2,B_3,B_4 defined as follows:

- *Definition of* B_1*:* B_1 is the binary string of length $|D|-1$ consisting of $|D|-2$ zeroes followed by a one.
- *Definition of* B_2*:* B_2 is the binary string of length $2|D|-2$ consisting of exactly $|D|-2$ ones, where the positions of the ones indicate the positions in $S(D)$ where the first left-to-right appearances of the elements of $V^*(D)$ occur.
- *Definition of* B_3*:* B_3 is the binary word consisting of alternate runs of ones and zeroes, with the first run of length f_1, the next run of length f_2, and so forth, with the last two runs of length $f_{|D|-2}$ and 1 respectively. The total number of runs in B_3 is $|D|-1$, and since $f_T>1$, the length of B_3 is $\leq 2|D|-3$.
- *Definition of* B_4*:* If $M(D)=1$, define B_4 to be the empty string. Assume $M(D)>1$. List the elements of $V^*(D)\cup\{T\}$ in the order $1,2,\cdots,|D|-2,T$, and then list the elements of $\mathcal{S}(D)'$ in the resulting lexicographical order. Let I be the integer such that $S(D)'$ is the I-th member of this list. B_4 is defined to be the expansion of integer $I-1$ into $\lceil\log_2 M(D)\rceil$ bits.

We discuss how D would be decoded from its codeword $\phi(D)$. Scanning $\phi(D)$ from left to right to find the first 1, B_1 is determined which yields $|D|$. B_2 is then determined by the fact that it is of length $2|D|-2$, and then B_3 is determined from the fact that it consists of $|D|-1$ runs. B_4 is then what is left over at the right end of $\phi(D)$. (Note that if instead of $\phi(D)$ one is given a longer binary word of which $\phi(D)$ is a prefix, knowledge of B_1 and B_3 allows one to compute the length $\lceil\log_2 M(D)\rceil$ of B_4 and thereby determine where word B_4 ends; this fact is what makes ϕ a prefix encoder on $\mathcal{D}$.) Knowledge of B_2 allows the decoder to fill in the positions in $S(D)$ corresponding to the 1's in B_2 with the symbols $1,2,\cdots,|D|-2$ left-to-right (they appear in this order due to the way in which the vertices of D were named in Sec. II). Knowledge of B_3 allows the decoder to determine the set $\mathcal{S}(D)'$ and then knowledge of B_4 allows the decoder to determine where $S(D)'$ lies in the listing of the $\mathcal{S}(D)'$ sequences. $S(D)'$ is then used by the decoder to fill in the remaining positions in $S(D)$. $|D|$ and $S(D)$ now being known, the determination of D from codeword $\phi(D)$ has been completed.

Example 4. Let t be the tree in Fig. 3 and let $D=D(t)$. We compute $\phi(D)$. From the rules of $D(t)$ in Fig. 3, we see that $|D|=8$ and

$$S(D)=(1,2,3,4,3,T,5,4,T,6,6,T,T,T),$$

$$f_1=f_2=f_5=1,\ f_3=f_4=f_6=2,\ f_T=5,$$

$$S(D)'=(3,T,4,T,6,T,T,T)$$

$$B_1=0000001,$$

$$B_2=11110010010000,$$

$$B_3=1011001001.$$

Listing the $8!/5!=336$ members of $\mathcal{S}(D)'$ is lexicographical order, one sees that $S(D)'$ is in position $I=14$ in this list; alternatively, the method of Cover [3] can be used. We conclude that B_4 is the $\lceil\log_2 336\rceil=9$ bit binary expansion of integer $14-1=13$; that is,

$$B_4=000001101.$$

The codeword $\phi(D)=B_1B_2B_3B_4$ is of length $7+14+10+9=40$.

Definition. Given $D\in\mathcal{D}$, let p_D be the first-order empirical probability distribution of the sequence $S(D)'$, defined on the alphabet A of $S(D)'$. Let $H(p_D)$ be the Shannon entropy of p_D, that is

$$H(p_D)\triangleq\sum_{v\in A}-p_D(v)\log_2 p_D(v).$$

Lemma 1. The encoder ϕ on $\mathcal{D}$ satisfies

$$L[\phi(D)]\leq 5(|D|-1)+|D|H(p_D),\ \ D\in\mathcal{D}. \tag{8}$$

Proof. If $|D|=2$, (8) trivially follows from the facts that $L[\phi(D)]=1$ and $H(p_D)=0$. Assume $|D|>2$. From the relationships

$$L[\phi(D)]=\sum_{i=1}^{4}L[B_i]=3(|D|-1)+L[B_3]+\lceil\log_2 M(D)\rceil,$$

$$L[B_3] \le 2|D| - 3,$$

$$\lceil \log_2 M(D) \rceil \le \log_2 M(D) + 1,$$

we obtain

$$L[\phi(D)] \le 5(|D|-1) + \log_2 M(D).$$

Since $\mathcal{S}(D)'$ is a type class of sequences of length $|D|$ in the sense of Chapter 2 of [4], Lemma 2.3 of [4] tells us that

$$\log_2 M(D) \le |D| H(p_D).$$

Combining the two previous inequalities, (8) results.

IV. Proof of Theorem 1

Fix $\lambda \in \Lambda$. We prove (3) holds for some constant $C(\lambda)$, where ϕ^* is the two-step encoder on $\mathcal{T}$ that was defined in Sec. III. Fix $t \in \mathcal{T}$ and let $D = D(t)$. As in Fig. 4, one can prune edges from t until one is left with a binary tree $t^\dagger$ that is a subtree of t rooted at the root of t such that

- There are $|D|$ leaf vertices of $t^\dagger$.
- $\{t(v) : v \in V^1(t^\dagger)\} = \{t_0, t_1, \cdots, t_{|D|-2}\}$.

It follows that there is an ordering $v_1, v_2, \cdots, v_{|D|}$ of the $|D|$ leaves of $t^\dagger$ so that the sequence $s = (t(v_1), t(v_2), \cdots, t(v_{|D|}))$ is simply a re-labeling of the entries of the sequence $S(D)'$. Thus, letting p^* be the first-order empirical probability distribution of s, the Shannon entropy $H(p^*)$ of p^* must coincide with $H(p_D)$. Thus,

$$|D|H(p_D) = |D|H(p^*) = \sum_{i=1}^{|D|} -\log_2 p^*(t(v_i)).$$

By Shannon's inequality, if q is any probability distribution on $\mathcal{T}^*$, we have

$$\sum_{i=1}^{|D|} -\log_2 p^*(t(v_i)) \le \sum_{i=1}^{|D|} -\log_2 q(t(v_i)). \tag{9}$$

To finish the proof, we just have to select q in the appropriate way. Define

$$M_j \stackrel{\Delta}{=} \sum_{u \in \mathcal{T}_j} \lambda(u), \;\; j \ge 1.$$

There is a unique real number $B > 1/2$ such that

$$q(u) \stackrel{\Delta}{=} B M_j^{-1} |u|^{-2} \lambda(u), \;\; u \in \mathcal{T}_j, \;\; j \ge 1 \tag{10}$$

defines a probability distribution on $\mathcal{T}^*$. We then have

$$\begin{aligned}\sum_{i=1}^{|D|} -\log_2 q(t(v_i)) &= |D|(-\log_2 B) + 2Q_1 + Q_2 + Q_3 \\ &\le |D| + 2Q_1 + Q_2 + Q_3,\end{aligned}$$

where

$$\begin{aligned} Q_1 &= \sum_{i=1}^{|D|} \log_2 |t(v_i)|, \\ Q_2 &= -\sum_{i=1}^{|D|} \log_2 \lambda(t(v_i)), \\ Q_3 &= \sum_{i=1}^{|D|} \log_2 M_{|t(v_i)|}. \end{aligned}$$

By concavity of the logarithm function and (1)-(2), we obtain the bounds

$$\begin{aligned} Q_1 &\le |D| \log_2 \left(\frac{\sum_{i=1}^{|D|} |t(v_i)|}{|D|} \right) = -|t| r(t) \log_2 r(t), \\ Q_2 &\le -\log_2 \lambda(t), \\ Q_3 &\le K(\lambda) Q_1. \end{aligned}$$

Incorporating these bounds in the preceding and employing (8), we obtain the pointwise redundancy bound

$$|t|^{-1}\{L[\phi^*(t)] + \lambda(t)\} \le 6r(t) - (K(\lambda)+2) r(t) \log_2 r(t),$$

whence (3) holds with $C(\lambda) = 2K(\lambda) + 16$.

Final Remark. The paper [9] gives full details on the results which were summarized in the present work, plus other results.

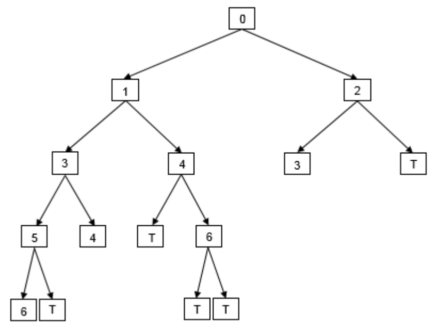


Fig. 4: Pruning of Fig. 3 tree used in Theorem 1 proof

Acknowledgement

The research of E.-H. Yang in this work is supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant RGPIN203035-11, and by the Canada Research Chairs Program.

References

[1] G. Busatto, M. Lohrey, and S. Maneth, "Grammar-Based Tree Compression," Technical Report IC/2004/80, EPFL, Lausanne, Switzerland, 17 pp., 2004.

[2] Y. Choi and W. Szpankowski, "Compression of graphical structures: fundamental limits, algorithms, and experiments," *IEEE Trans. Inform. Theory*, Vol. 58, pp. 620–638, 2012.

[3] T. Cover, "Enumerative source encoding," *IEEE Trans. Inform. Theory*, Vol. 19, pp. 73–77, 1973.

[4] I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems.* Akadémiai Kiadó, Budapest, Hungary, 1981.

[5] J. Kieffer and E.-H. Yang, "Grammar-Based Codes: A New Class of Universal Lossless Source Codes," *IEEE Trans. Inform. Theory*, Vol. 46, pp. 737–754, 2000.

[6] J. Kieffer, E.-H. Yang, and W. Szpankowski "Structural complexity of random binary trees," *Proc. 2009 IEEE International Symposium on Inform. Theory (Seoul, Korea)*, pp. 635-639, 2009.

[7] S. Maneth and G. Busatto, "Tree transducers and tree compressions," *Proc. Foundations of Software Science and Computation Structures (FOSSACS 2004)*, Lecture Notes in Computer Science, Vol. 2987, pp. 363–377, Springer-Verlag, 2004.

[8] R. Stanley, *Enumerative Combinatorics, Vol. 2.* Cambridge University Press, Cambridge, UK, 1999.

[9] J. Zhang, E.-H. Yang, and J. Kieffer, "A Universal Grammar-Based Code For Lossless Compression of Binary Trees," arXiv:1304.7392.