Universal Tree Source Coding Using Grammar-Based Compression

Danny Hucke and Markus Lohrey
University of Siegen, Germany

***Abstract*—We apply so-called tree straight-line programs to the problem of universal source coding for binary trees. We derive an upper bound on the maximal pointwise redundancy (or worst-case redundancy) that improve previous bounds on the average case redundancy obtained by Zhang, Yang, and Kieffer using directed acyclic graphs. Using this, we obtain universal codes for new classes of tree sources.**

I. Introduction

Universal source coding for finite sequences over a finite alphabet Σ (i.e., strings over Σ) is a well-established topic of information theory. Its goal is to find prefix-free lossless codes that are universal (or optimal) for classes of information sources. Kieffer and Yang developed grammar-based codes that are universal for the class of finite state sources [8]. Grammar-based compression works in two steps: In a first step, from a given input string $w \in \Sigma^*$ a context-free grammar $\mathcal{G}_w$ that produces only the string w is computed. Context-free grammars that produce exactly one string are also known as *straight-line programs*, briefly SLPs, and are currently an active topic in text compression and algorithmics on compressed texts, see [10] for a survey. In a second step, the SLP $\mathcal{G}_w$ is encoded by a binary string $B(\mathcal{G}_w)$. There exist several algorithms that compute from a given input string w of length n an SLP $\mathcal{G}_w$ of size $O(n/\log n)$ (the size of an SLP is the total number of symbols in all right-hand sides of the grammar) [8]; the best known example is probably the LZ78 algorithm [14]. By combining any of these algorithms with the binary encoder B for SLPs from [8], one obtains a grammar-based encoder $E : \Sigma^* \to \{0,1\}^*$, whose worst-case redundancy for input strings of length n is bounded by $O(\log\log n/\log n)$ for every finite state information source over the alphabet Σ. Here, the worst-case redundancy for strings of length n is defined as the maximum of $n^{-1} \cdot (|E(w)| + \log_2 P(w))$ taken over all words $w \in \Sigma^n$ with $P(w) > 0$, where $P(w)$ is the probability that the finite state information source emits w. Thus, the worst-case redundancy measures the maximal additive deviation of the code length from the self information, normalized by the length of the source string.

Over the last few years, we have seen increasing efforts aiming to extend universal source coding to structured data like trees [9], [12], [13] and graphs [1], [7]. In this paper, we are concerned with trees. In their recent paper [13], Zhang, Yang, and Kieffer started to extend grammar-based source coding from strings to binary trees. For this, they first represent a binary tree t by its *minimal directed acyclic graph* $\mathcal{D}_t$ (the minimal DAG of t). This is the directed acyclic graph obtained by removing multiple occurrences of the same subtree from t. In a second step, $\mathcal{D}_t$ is encoded by a binary string $B(\mathcal{D}_t)$; this step is similar to the binary coding of SLPs from [8]. Combining both steps yields a tree encoder $E_{\text{dag}} : \mathcal{T} \to \{0,1\}^*$, where $\mathcal{T}$ is the set of all binary trees. In order to define universality of such a tree encoder, the classical notion of an information source on finite sequences is replaced in [13] by the notion of a tree source. This is a collection of probability distributions $(P_n)_{n\in\mathbb{N}}$, where every P_n is a distribution on a finite non-empty subset $F_n \subseteq \mathcal{T}$, and these sets partition $\mathcal{T}$ (see also Sec. II-B). Two classes of tree sources are considered in [13]: leaf-centric sources (F_n is the set of all binary trees with n leaves) and depth-centric sources (F_n is the set of all binary trees of depth n). Then, two properties on binary tree sources are introduced in [13]: the domination property (see Sec. III, where it is called the weak domination property) and the representation ratio negligibility property. The latter states that $\sum_{t\in F_n} P_n(t) \cdot |\mathcal{D}_t|/|t|$ (the average compression ratio achieved by the minimal DAG) converges to zero for $n \to \infty$, where the size $|t|$ of the binary tree is defined as its number of leaves. The main result of [13] states that for every tree source $(P_n)_{n\in\mathbb{N}}$ satisfying the domination property and the representation ratio negligibility property the *average case redundancy*

$$\sum_{t\in F_n, P_n(t)>0} |t|^{-1} \cdot (|E_{\text{dag}}(t)| + \log_2 P_n(t)) \cdot P_n(t)$$

converges to zero for $n \to \infty$. Finally, two classes of tree sources having the domination property and the representation ratio negligibility property are presented in [13]. One is a class of leaf centric sources, the other one is a class of depth centric sources. Both sources have the property that every tree with a non-zero probability is balanced in a certain sense, the precise definitions can be found in Sec. III-C. As a first contribution, we show that for these sources not only the average case redundancy but also the worst-case redundancy

$$\max_{t\in F_n, P_n(t)>0} |t|^{-1} \cdot (|E_{\text{dag}}(t)| + \log_2 P_n(t)) \qquad (1)$$

converges to zero for $n \to \infty$. More precisely, we show that (1) is bounded by $O(\log\log n/\log n)$ (resp., $O((\log\log n)^2/\log n)$) for the presented class of leaf-centric tree sources (resp., depth-centric tree sources). To prove this, we use results from [5], [6] according to which the minimal DAG of a suitably balanced binary tree of size n is bounded by $O(n/\log n)$, respectively $O(n \cdot \log\log n/\log n)$.

Our second main contribution is the application of *tree straight-line programs*, briefly TSLPs, for universal tree coding. A TSLP is a context-free tree grammar that produces exactly one tree, see Sec. II-C for the precise definition and [11] for a survey. TSLPs can be viewed as the proper generalization of SLPs for trees. Whereas DAGs only have the ability to share repeated subtrees of a tree, TSLPs can also share repeated tree patterns with a hole (so-called contexts). In [5], the authors presented a linear time algorithm that computes for a given binary tree t of size n a TSLP $\mathcal{G}_t$ of size $O(n/\log n)$. This shows the main advantage of TSLPs over DAGs: There exist trees of any size n for which the minimal DAG has size n as well. In Sec. IV-B we define a binary encoding B of TSLPs similar to the ones for SLPs [8] and DAGs [13]. We then consider the combined tree encoder $E_{\text{tslp}} : \mathcal{T} \to \{0,1\}^*$ with $E_{\text{tslp}}(t) = B(\mathcal{G}_t)$, and prove that its worst-case redundancy (defined as in (1) with E_{dag} replaced by E_{tslp}) is bounded by $O(\log\log n/\log n)$ for every tree source that satisfies the *strong domination property* defined in Sec. IV-C. The strong domination property is a strengthening of the domination property from [13], and this is what we have to pay extra for our TSLP-based encoding in contrast to the DAG-based encoding from [13]. On the other hand, the TSLP-based encoding has two main advantages over the DAG-based encoding: (i) The representation ratio negligibility property from [13] is no longer needed, and (ii) we get bounds on the worst-case redundancy instead of the average case redundancy. Both advantages are based on the fact that the grammar-based compressor from [5] computes a TSLP of size $O(n/\log n)$ for every binary tree of size n. We conclude the paper with the presentation of two natural classes of leaf-centric and depth-centric tree sources having the strong domination property. These classes are orthogonal to the classes from [13].

For a full version of the paper see [4].

II. Preliminaries

In this section, we introduce basic definitions concerning information theory (Sec. II-A), binary trees (Sec. II-B) and tree straight-line programs (Sec. II-C). The latter are our key formalism for compressing binary trees. With $\mathbb{N}$ we denote the natural numbers including 0. We use the standard O-notation and for a constant b we write $O(\log n)$ instead of $O(\log_b n)$.

A. Empirical distributions and empirical entropy

Let $\overline{a} = (a_1, a_2, \dots, a_n)$ be a tuple of elements that are from some (not necessarily finite) set A. The *empirical distribution* $p_{\overline{a}} : \{a_1, a_2, \dots, a_n\} \to \mathbb{R}$ of $\overline{a}$ is defined by $p_{\overline{a}}(a) = n^{-1} \cdot |\{i \mid 1 \le i \le n,\ a_i = a\}|$. We use this definition also for words over some alphabet by identifying a word $w = a_1 a_2 \cdots a_n$ with the tuple $(a_1, a_2, \dots, a_n)$. The *unnormalized empirical entropy* of $\overline{a}$ is $H(\overline{a}) := -\sum_{i=1}^{n} \log p_{\overline{a}}(a_i)$.

B. Trees, tree sources, and tree compressors

With $\mathcal{T}$ we denote the set of all binary trees. We identify $\mathcal{T}$ with the set of terms that are built from the binary symbol f and the constant a. Formally, $\mathcal{T}$ is the smallest set such that $a \in \mathcal{T}$ and if $t_1, t_2 \in \mathcal{T}$ then also $f(t_1, t_2) \in \mathcal{T}$. With $|t|$ we denote the number of occurrences of the constant a in t. This is the number of leaves of t. Let $\mathcal{T}_n = \{t \in \mathcal{T} \mid |t| = n\}$ for $n \ge 1$. The depth $d(t)$ of the tree t is recursively defined by $d(a) = 0$ and $d(f(t_1, t_2)) = \max\{d(t_1), d(t_2)\} + 1$. Let $\mathcal{T}^d = \{t \in \mathcal{T} \mid d(t) = d\}$ for $d \in \mathbb{N}$.

Occasionally, we will consider a binary tree t as a graph with nodes and edges in the usual way. Note that a tree $t \in \mathcal{T}_n$ has $2n-1$ nodes in total: n leaves and $n-1$ internal nodes. For a node v we write $t[v]$ for the *subtree* rooted at v in t.

A *context* is a binary tree t, where exactly one leaf is labelled with the special symbol x (called the *parameter*); all other leaves are labelled with a. For a context t we define $|t|$ to be the number of a-labelled leaves of t (which is the number of leaves of t minus 1). We denote with $\mathcal{C}$ the set of all contexts and define $\mathcal{C}_n = \{t \in \mathcal{C} \mid |t| = n\}$ for $n \in \mathbb{N}$. For a tree or context $t \in \mathcal{T} \cup \mathcal{C}$ and a context $s \in \mathcal{C}$, we denote by $s(t)$ the tree or context which results from s by replacing the parameter x by t. For example $s = f(a, x)$ and $t = f(a, a)$ yields $s(t) = f(a, f(a, a))$. The depth $d(t)$ of a context $t \in \mathcal{C}$ is defined as the depth of the tree $t(a)$.

A tree source is a pair $((\mathcal{F}_i)_{i \in \mathbb{N}}, P)$ such that:

- $\mathcal{F}_i \subseteq \mathcal{T}$ is non-empty and finite for every $i \ge 0$,
- $\mathcal{F}_i \cap \mathcal{F}_j = \emptyset$ for $i \ne j$ and $\bigcup_{i \ge 0} \mathcal{F}_i = \mathcal{T}$, i.e., the sets $\mathcal{F}_i$ form a partition of $\mathcal{T}$,
- $P : \mathcal{T} \to [0,1]$ and $\sum_{t \in \mathcal{F}_i} P(t) = 1$ for every $i \ge 0$, i.e., P restricted to $\mathcal{F}_i$ is a probability distribution.

In this paper, we consider only two cases for the partition $(\mathcal{F}_i)_{i \in \mathbb{N}}$: either $\mathcal{F}_i = \mathcal{T}_{i+1}$ for all $i \in \mathbb{N}$ (note that there is no tree of size 0) or $\mathcal{F}_i = \mathcal{T}^i$ for all $i \in \mathbb{N}$. Tree sources of the former (resp., latter) type are called *leaf-centric* (resp., *depth-centric*). More specifically, we follow [13] to specify a leaf-centric (resp. depth-centric) tree source as follows: Let Σ_{leaf} be the set of all functions $\sigma : (\mathbb{N}\backslash\{0\}) \times (\mathbb{N}\backslash\{0\}) \to [0,1]$ such that for all $n \ge 2$:

$$\sum_{i,j \ge 1,\ i+j=n} \sigma(i,j) = 1. \tag{2}$$

Moreover, let Σ_{depth} be the set of all mappings $\sigma : \mathbb{N} \times \mathbb{N} \to [0,1]$ such that for all $n \ge 1$:

$$\sum_{i,j \ge 0,\ \max(i,j)=n-1} \sigma(i,j) = 1. \tag{3}$$

For $\sigma \in \Sigma_{\text{leaf}}$ we define $P_\sigma : \mathcal{T} \to [0,1]$ inductively by

$$P_\sigma(a) = 1 \text{ and } P_\sigma(f(s,t)) = \sigma(|s|, |t|) \cdot P_\sigma(s) \cdot P_\sigma(t). \tag{4}$$

We have $\sum_{t \in \mathcal{T}_n} P_\sigma(t) = 1$ and thus $\mathcal{S}^l_\sigma := ((\mathcal{T}_i)_{i \ge 1}, P_\sigma)$ is a leaf-centric tree source.

For $\sigma \in \Sigma_{\text{depth}}$, we define $P_\sigma : \mathcal{T} \to [0,1]$ by

$$P_\sigma(a) = 1,\ P_\sigma(f(s,t)) = \sigma(d(s), d(t)) \cdot P_\sigma(s) \cdot P_\sigma(t). \tag{5}$$

We have $\sum_{t \in \mathcal{T}^n} P_\sigma(t) = 1$. Thus, $\mathcal{S}^d_\sigma := ((\mathcal{T}^i)_{i \ge 0}, P_\sigma)$ is a depth-centric tree source.

A *tree encoder* is an injective mapping $E : \mathcal{T} \to \{0,1\}^*$ whose range $E(\mathcal{T})$ is prefix-free, i.e., there do not exist $t, t' \in$

$\mathcal{T}$ with $t \neq t'$ such that $E(t)$ is a prefix of $E(t')$. We define the *worst-case redundancy* (also known as the *maximal pointwise redundancy*) of E w.r.t. the tree source $\mathcal{S} = ((\mathcal{F}_i)_{i\in\mathbb{N}}, P)$ as the mapping $i \mapsto R(E, \mathcal{S}, i)$ ($i \in \mathbb{N}$) with

$$R(E,\mathcal{S},i) := \max_{t\in\mathcal{F}_i, P(t)>0} |t|^{-1} \cdot (|E(t)| + \log_2 P(t))$$

C. Tree straight-line programs

We now introduce tree straight-line programs. Let V be a finite set of *nonterminals* disjoint from $\{f, a, x\}$. Each symbol $A \in V$ has an associated rank 0 or 1. We use V_0 (resp., V_1) for the set of nonterminals of rank 0 (resp. of rank 1) and we assume that $V_0 \neq \emptyset$. The idea is that nonterminals from V_0 (resp., V_1) derive to trees from $\mathcal{T}$ (resp., contexts from $\mathcal{C}$). We denote by $\mathcal{T}_V$ the set of trees over $\{f, a\} \cup V$, i.e. each node in a tree $t \in \mathcal{T}_V$ is labelled with a symbol from $\{f, a\} \cup V$ and the number of children of a node corresponds to the rank of its label. With $\mathcal{C}_V$ we denote the corresponding set of all contexts, i.e., the set of trees over $\{f, a, x\} \cup V$, where the parameter symbol x occurs exactly once and at a leaf position. Clearly, $\mathcal{T} \subset \mathcal{T}_V$ and $\mathcal{C} \subset \mathcal{C}_V$. A *tree straight-line program*, or short *TSLP*, is a tuple $\mathcal{G} = (V, A_0, r)$, where $A_0 \in V_0$ is the start nonterminal and $r : V \to (\mathcal{T}_V \cup \mathcal{C}_V)$ is the function which assigns each nonterminal its right-hand side. It is required that if $A \in V_0$ (resp., $A \in V_1$), then $r(A) \in \mathcal{T}_V$ (resp., $r(A) \in \mathcal{C}_V$). Furthermore, the binary relation $\{(A, B) \in V \times V \mid B \text{ is a label in } r(A)\}$ needs to be acyclic. These conditions ensure that exactly one tree is derived from the start nonterminal A_0 by using the rules $A \to r(A)$ for $A \in V$. Formally, we define $\mathsf{val}_{\mathcal{G}}(t) \in \mathcal{T}$ for $t \in \mathcal{T}_V$ and $\mathsf{val}_{\mathcal{G}}(t) \in \mathcal{C}$ for $t \in \mathcal{C}_V$ inductively:

- $\mathsf{val}_{\mathcal{G}}(a) = a$ and $\mathsf{val}_{\mathcal{G}}(x) = x$
- $\mathsf{val}_{\mathcal{G}}(f(t_1, t_2)) = f(\mathsf{val}_{\mathcal{G}}(t_1), \mathsf{val}_{\mathcal{G}}(t_2))$ for $f(t_1, t_2) \in \mathcal{T}_V \cup \mathcal{C}_V$
- $\mathsf{val}_{\mathcal{G}}(A) = \mathsf{val}_{\mathcal{G}}(r(A))$ for $A \in V_0$
- $\mathsf{val}_{\mathcal{G}}(A(s)) = t'(\mathsf{val}_{\mathcal{G}}(s))$ for $A \in V_1$, $t' = \mathsf{val}_{\mathcal{G}}(r(A)) \in \mathcal{C}$, and $s \in \mathcal{T}_V \cup \mathcal{C}_V$.

The tree defined by $\mathcal{G}$ is $\mathsf{val}(\mathcal{G}) := \mathsf{val}_{\mathcal{G}}(A_0) \in \mathcal{T}$. Moreover, for $A \in V_1$ we also write $\mathsf{val}_{\mathcal{G}}(A)$ for $\mathsf{val}_{\mathcal{G}}(A(x))$.

Example 1. *Let $\mathcal{G} = (\{A_0, A_1, A_2\}, A_0, r)$ be a TSLP with $A_0, A_1 \in V_0, A_2 \in V_1$, $r(A_0) = f(A_1, A_2(a))$, $r(A_1) = A_2(A_2(a))$, $r(A_2) = f(x, a)$. We get $\mathsf{val}_{\mathcal{G}}(A_2) = f(x, a)$, $\mathsf{val}_{\mathcal{G}}(A_1) = f(f(a, a), a)$ and $\mathsf{val}(\mathcal{G}) = \mathsf{val}_{\mathcal{G}}(A_0) = f(f(f(a, a), a), f(a, a))$.*

In this paper, we will consider two classes of syntactically restricted TSLPs: (i) DAGs (directed acyclic graphs) and (ii) TSLPs in normal form. Let us start with the former; normal form TSLPs will be introduced in Sec. IV-A.

III. Tree compression with DAGs

In this section we sharpen some of the results from [13], where universal source coding of binary trees using minimal DAGs (directed acyclic graphs) is investigated. In [13], only bounds on the average redundancy for certain classes of tree sources were shown. Here we extend these bounds (for the same classes of tree sources) to the worst-case redundancy.

A. Directed acyclic graphs (DAGs)

A DAG is a TSLP $\mathcal{D} = (V, A_0, r)$ such that $V = \{A_0, A_1, \ldots, A_{n-1}\}$ for some $n \in \mathbb{N}$, $n \geq 1$, $V = V_0$ (i.e., all nonterminals have rank 0), and for every $A_i \in V$, the right-hand side $r(A_i)$ is of the form $f(\alpha_1, \alpha_2)$ with $\alpha_1, \alpha_2 \in \{a, A_{i+1}, \ldots, A_{n-1}\}$. Its size is $|\mathcal{D}| = n + 1$. Note that a TSLP of this form generates a tree with at least two leaves. In order to include the tree a with a single leaf, we also allow the TSLP $\mathcal{G}_a = (\{A_0\}, A_0, A_0 \mapsto a)$ of size 1.

In contrast to general TSLPs, every binary tree t has a unique (up to renaming of nonterminals) minimal DAG $\mathcal{D}_t$, whose size is the number of different (pairwise non-isomorphic) subtrees of t. The idea is to introduce for every subtree $f(t_1, t_2)$ of size at least two a nonterminal A_i with $r(A_i) = f(\alpha_1, \alpha_2)$, where $\alpha_i = a$ if $t_i = a$ and α_i is the nonterminal corresponding to the subtree t_i if $|t_i| \geq 2$. We will only use this minimal DAG $\mathcal{D}_t$ in the sequel. The following example shows that in the worst-case, the size of the minimal DAG is not smaller than the size of the tree.

Example 2. *Let $t_n = f(f(f(\cdots f(a, a), \cdots a), a), a) \in \mathcal{T}_{n+1}$, where f occurs n times. The minimal DAG of t_n is $(\{A_0, \ldots, A_{n-1}\}, A_0, r_n)$, where $r_n(A_i) = f(A_{i+1}, a)$ for $0 \leq i \leq n-2$ and $r_n(A_{n-1}) = f(a, a)$ and its size is $n + 1$.*

B. Universal source coding with DAGs

The following property was introduced in [13], where it is called the domination property (later, we will introduce a strong domination property): A tree source $((\mathcal{F}_i)_{i\in\mathbb{N}}, P)$ (as defined in Sec. II-B) has the *weak domination property* if there is a mapping $\lambda : \mathcal{T} \to \mathbb{R}_{>0}$ such that:

- $\lambda(t) \geq P(t)$ for every $t \in \mathcal{T}$,
- $\lambda(f(s, t)) \leq \lambda(s) \cdot \lambda(t)$ for all $s, t \in \mathcal{T}$, and
- there are constants c_1, c_2 such that $\sum_{t\in\mathcal{T}_n} \lambda(t) \leq c_1 \cdot n^{c_2}$ for all $n \geq 1$.

In [13], the authors define a binary encoding $B(\mathcal{D}_t) \in \{0, 1\}^*$, such that $B(\mathcal{D}_t)$ is not a prefix of $B(\mathcal{D}_{t'})$ for all binary trees t, t' with $t \neq t'$. The precise definition of $B(\mathcal{D}_t)$ is not important for us; all we need is the following bound from [13, Thm. 2], where $E_{\text{dag}} : \mathcal{T} \to \{0, 1\}^*$ is the tree encoder with $E_{\text{dag}}(t) = B(\mathcal{D}_t)$, $((\mathcal{F}_i)_{i\in\mathbb{N}}, P)$ is a tree source with the weak domination property, and $t \in \mathcal{T}_n$ ($n \geq 2$) with $P(t) > 0$:

$$\begin{aligned}&\frac{1}{n} \cdot (|E_{\text{dag}}(t)| + \log_2(P(t))) \\ &\leq O(|\mathcal{D}_t|/n) + O(|\mathcal{D}_t|/n \cdot \log_2(n/|\mathcal{D}_t|)).\end{aligned} \quad (6)$$

This bound is used in [13] to show that for certain leaf-centric and depth-centric tree sources the encoding E_{dag} is universal in the sense that the average redundancy converges to zero. In the next section, we show that for the same tree sources already the worst-case redundancy converges to zero.

C. Tree sources with the weak domination property

The following result is implicitly shown in [13].

Lemma 1. *For every $\sigma \in \Sigma_{leaf}$ (resp., $\sigma \in \Sigma_{depth}$) the leaf-centric (resp., depth-centric) tree source $\mathcal{S}^l_\sigma$ (resp., $\mathcal{S}^d_\sigma$) has the weak domination property.*

We say that $\sigma \in \Sigma_{\text{leaf}}$ is *leaf-balanced* if there exists a constant c such that $(i+j)/\min\{i,j\} \le c$ for all $(i,j) \in (\mathbb{N} \setminus \{0\}) \times (\mathbb{N} \setminus \{0\})$ with $\sigma(i,j) > 0$. In [13] it is shown that for a leaf-balanced $\sigma \in \Sigma_{\text{leaf}}$, the tree source $\mathcal{S}^l_\sigma$ has the property that the average compression ratio achieved by the minimal DAG (formally, $\sum_{t \in \mathcal{T}_n} P_\sigma(t) \cdot |\mathcal{D}_t|/n$) converges to zero for $n \to \infty$. Using a result from [5], we can show the following stronger property.

Lemma 2. *For every leaf-balanced mapping $\sigma \in \Sigma_{leaf}$, there exists a constant α such that for every binary tree $t \in \mathcal{T}_n$ with $P_\sigma(t) > 0$ we have $|\mathcal{D}_t| \le \alpha \cdot n/\log_2 n$.*

Lemma 2 and the bound (6) directly yield:

Corollary 1. *If the mapping $\sigma \in \Sigma_{leaf}$ is leaf-balanced, then $R(E_{dag}, \mathcal{S}^l_\sigma, i) \le O(\log\log i/\log i)$.*

We say $\sigma \in \Sigma_{\text{depth}}$ is *depth-balanced* if there exists a constant c such that $|i-j| \le c$ for all $(i,j) \in \mathbb{N} \times \mathbb{N}$ with $\sigma(i,j) > 0$. In [13], the authors define a condition on σ that is slightly stronger than depth-balancedness, and show that for every such σ, the average compression ratio achieved by the minimal DAG converges to zero. Using results from [5], [6], we can show the following stronger property:

Lemma 3. *For every depth-balanced mapping $\sigma \in \Sigma_{depth}$ there exists a constant α such that for every binary tree $t \in \mathcal{T}_n$ with $P_\sigma(t) > 0$ we have $|\mathcal{D}_t| \le \alpha \cdot n \cdot \log_2(\log_2 n)/\log_2 n$.*

Lemma 3 and the bound (6) yield:

Corollary 2. *If the mapping $\sigma \in \Sigma_{depth}$ is depth-balanced, then $R(E_{dag}, \mathcal{S}^d_\sigma, i) \le O((\log\log i)^2/\log i)$.*

IV. Tree compression with TSLPs

In this section, we will use general TSLPs for the compression of binary trees. The limitations of DAGs for universal source coding can be best seen for a tree source $((\mathcal{F}_i)_{i\in\mathbb{N}}, P)$ such that $P(t) > 0$ for all $t \in \mathcal{T}$. Example 2 shows that for every $n \ge 1$, there is a tree $t \in \mathcal{T}_n$ with $|\mathcal{D}_t| = n$. In that case, the bound (6) cannot be used to show that the worst-case redundancy converges to zero.

A. TSLPs in normal form

A TSLP $\mathcal{G} = (V, A_0, r)$ is in *normal form* if the following conditions hold:

- $V = \{A_0, A_1, \dots, A_{n-1}\}$ for some $n \in \mathbb{N}$, $n \ge 1$.
- For every $A_i \in V_0$, the right-hand side $r(A_i)$ is a term of the form $A_j(\alpha)$, where $A_j \in V_1$ and $\alpha \in V_0 \cup \{a\}$.
- For every $A_i \in V_1$ the right-hand side $r(A_i)$ is a term of the form $A_j(A_k(x))$, $f(\alpha, x)$, or $f(x,\alpha)$, where $A_j, A_k \in V_1$ and $\alpha \in V_0 \cup \{a\}$.
- For every $A_i \in V$ define $\rho(A_i) \in (V \cup \{a\})^*$ as
$$\rho(A_i) := \begin{cases} A_j\alpha & \text{if } r(A_i) = A_j(\alpha) \\ A_jA_k & \text{if } r(A_i) = A_j(A_k(x)) \\ \alpha & \text{if } r(A_i) = f(\alpha, x) \text{ or } f(x,\alpha). \end{cases}$$
Let $\rho_{\mathcal{G}} := \rho(A_0)\cdots\rho(A_{n-1}) \in \{a, A_1, \dots, A_{n-1}\}^*$. Then we require that $\rho_{\mathcal{G}}$ is of the form $\rho_{\mathcal{G}} = A_1u_1 \cdots A_{n-1}u_{n-1}$ with $u_i \in \{a, A_1, A_2, \dots, A_i\}^*$.
- $\mathsf{val}_{\mathcal{G}}(A_i) \ne \mathsf{val}_{\mathcal{G}}(A_j)$ for $i \ne j$

As for DAGs we allow the TSLP $\mathcal{G}_a = (\{A_0\}, A_0, A_0 \mapsto a)$ in order to get the tree a. In this case, we set $\rho_{\mathcal{G}_a} = \rho(A_0) = a$.

Let $\mathcal{G} = (V, A_0, r)$ be a TSLP in normal form with $V = \{A_0, \dots, A_{n-1}\}$. We define the size of $\mathcal{G}$ as $|\mathcal{G}| = |\rho_{\mathcal{G}}|$. This is the total number of occurrences of symbols from $V \cup \{a\}$ in all right-hand sides of $\mathcal{G}$. Let $\omega_{\mathcal{G}}$ be the word obtained from $\rho_{\mathcal{G}}$ by removing for every $1 \le i \le n-1$ the first occurrence of A_i from $\rho_{\mathcal{G}}$. Thus, if $\rho_{\mathcal{G}} = A_1u_1A_2u_2\cdots A_{n-1}u_{n-1}$ with $u_i \in \{a, A_1, A_2, \dots, A_i\}^*$, then $\omega_{\mathcal{G}} = u_1u_2\cdots u_{n-1}$. The *entropy* $H(\mathcal{G})$ of the normal form TSLP $\mathcal{G}$ is defined as the empirical unnormalized entropy of the word $\omega_{\mathcal{G}}$: $H(\mathcal{G}) := H(\omega_{\mathcal{G}})$.

Example 3. *Let $\mathcal{G} = (\{A_0, A_1, A_2, A_3, A_4\}, A_0, r)$ be the normal form TSLP with $A_0, A_2, A_3 \in V_0$, $A_1, A_4 \in V_1$ and $r(A_0) = A_1(A_2)$, $r(A_1) = f(x, A_3)$, $r(A_2) = A_4(A_3)$, $r(A_3) = A_4(a)$, $r(A_4) = f(x,a)$. We have $\mathsf{val}(\mathcal{G}) = f(f(f(a,a),a),f(a,a))$, $\rho_{\mathcal{G}} = A_1A_2A_3A_4A_3A_4aa$, $|\mathcal{G}| = 8$ and $\omega_{\mathcal{G}} = A_3A_4aa$.*

A *grammar-based tree compressor* is an algorithm ψ that produces for a given tree $t \in \mathcal{T}$ a TSLP $\mathcal{G}_t$ in normal form. The *compression ratio* of ψ is the mapping $n \mapsto \gamma_\psi(n)$ with
$$\gamma_\psi(n) := \max_{t \in \mathcal{T}_n} |\mathcal{G}_t|/n.$$
Every TSLP can be transformed with a linear size increase into a normal form TSLP that derives the same tree. For example, the TSLP from Example 1 is transformed into the normal form TSLP described in Example 3. We will not use this fact, since all we need is the following theorem from [5]:

Theorem 1. *There exists a grammar-based compressor ψ (working in linear time) with $\gamma_\psi(n) \in O(1/\log n)$.*

B. Binary coding of TSLPs in normal form

In this section we fix a binary encoding for normal form TSLPs. This encoding is similar to the one for SLPs [8] and DAGs [13]. Let $\mathcal{G} = (V, A_0, r)$ be a TSLP in normal form with $n = |V|$ nonterminals. Let $m = |\mathcal{G}| = |\rho_{\mathcal{G}}|$ be the size of $\mathcal{G}$. We define the type $\tau(A_i) \in \{0,1,2,3\}$ of a nonterminal $A_i \in V$ as follows:
$$\tau(A_i) = \begin{cases} 0 & \text{if } \rho(A_i) \in V_1(V_0 \cup \{a\}) \\ 1 & \text{if } \rho(A_i) \in V_1V_1 \\ 2 & \text{if } \rho(A_i) = f(\alpha, x) \text{ for some } \alpha \in V_0 \cup \{a\} \\ 3 & \text{if } \rho(A_i) = f(x, \alpha) \text{ for some } \alpha \in V_0 \cup \{a\} \end{cases}$$
We define the binary word $B(\mathcal{G}) := w_0w_1w_2w_3w_4$, where $w_0 = 0^{n-1}1$ and $w_1, w_2, w_3, w_4 \in \{0,1\}^+$ are defined as follows: Let $w_1 = a_0b_0\cdots a_{n-1}b_{n-1}$, where a_jb_j

is the 2-bit binary encoding of $\tau(A_j)$. Next, let $\rho_\mathcal{G} = A_1 u_1 A_2 u_2 \cdots A_{n-1} u_{n-1}$ with $u_i \in \{a, A_1, \ldots, A_i\}^*$. Then $w_2 = 10^{|u_1|}10^{|u_2|} \cdots 10^{|u_{n-1}|}$. To define w_3, let $k_i = |\rho_\mathcal{G}|_{A_i} \geq 1$ $(1 \leq i \leq n-1)$ be the number of occurrences of the nonterminal A_i in the word $\rho_\mathcal{G}$. Then $w_3 = 0^{k_1-1}10^{k_2-1}1 \cdots 0^{k_{n-1}-1}1$. Finally, the word w_4 encodes the word $\omega_\mathcal{G}$ using enumerative encoding [2]: Every nonterminal A_i, $1 \leq i \leq n-1$, has $\eta(A_i) := k_i - 1$ occurrences in $\omega_\mathcal{G}$. The symbol a has $\eta(a) := m - (k_1 + \cdots + k_{n-1})$ many occurrences in $\omega_\mathcal{G}$. Let S be the set of words over the alphabet $\{a, A_1, \ldots, A_{n-1}\}$ with $\eta(a)$ occurrences of a and $\eta(A_i)$ occurrences of A_i for every $1 \leq i \leq n-1$. Hence,

$$|S| = (m - n + 1)! \,/\, (\eta(a)! \cdot \prod_{i=1}^{n-1} \eta(A_i)!) \qquad (7)$$

Let $v_0, v_1, \ldots, v_{|S|-1}$ be the lexicographic enumeration of the words from S w.r.t. the alphabet order $a, A_1, \ldots, A_{n-1}$. Then w_4 is the binary encoding of the unique i such that $\omega_\mathcal{G} = v_i$, where $|w_4| = \lceil \log_2 |S| \rceil$ (leading zeros are added to the binary encoding of i to obtain length $\lceil \log_2 |S| \rceil$).

Example 4. *Consider the normal from TSLP $\mathcal{G}$ from Example 3. We have $w_0 = 00001$, $w_1 = 0011000011$, $w_2 = 11110000$, $w_3 = 110101$. To compute w_4, note first that there are $|S| = 12$ words with two occurrences of a and one occurrence of A_3 and A_4. It follows that $|w_4| = \lceil \log_2(12) \rceil = 4$. Further, since the order of the alphabet is a, A_3, A_4, there are only three words in S (A_4A_3aa, A_4aA_3a and A_4aaA_3), which are lexicographically larger than $\omega_\mathcal{G} = A_3A_4aa$. Hence, $\omega_\mathcal{G} = v_8$ and $w_4 = 1000$.*

It is easy to show that the set of code words $B(\mathcal{G})$, where $\mathcal{G}$ ranges over all TSLPs in normal form, is a prefix code. Moreover, note that $|B(\mathcal{G})| \leq O(|\mathcal{G}|) + |w_4|$. By using the well-known bound on the code length of enumerative encoding [3, Thm. 11.1.3], we get the following lemma:

Lemma 4. *We have $|B(\mathcal{G})| \leq O(|\mathcal{G}|) + H(\mathcal{G})$.*

C. Universal source coding based on TSLPs in normal form

Let $((\mathcal{F}_i)_{i\in\mathbb{N}}, P)$ be a tree source as defined in Sec. II-B. We say that $((\mathcal{F}_i)_{i\in\mathbb{N}}, P)$ has the *strong domination property* if there exists a mapping $\lambda : \mathcal{T} \cup \mathcal{C} \to \mathbb{R}_{>0}$ such that:

(i) $\lambda(t) \geq P(t)$ for every $t \in \mathcal{T}$,
(ii) $\lambda(f(s,t)) \leq \lambda(s) \cdot \lambda(t)$ for all $s, t \in \mathcal{T}$,
(iii) $\lambda(s(t)) \leq \lambda(s) \cdot \lambda(t)$ for all $s \in \mathcal{C}$ and $t \in \mathcal{T}$, and
(iv) there are constants c_1, c_2 such that $\sum_{t\in\mathcal{T}_n\cup\mathcal{C}_n} \lambda(t) \leq c_1 \cdot n^{c_2}$ for all $n \geq 1$.

The proof of the next lemma combines ideas from [8], [13].

Lemma 5. *Assume that $((\mathcal{F}_i)_{i\in\mathbb{N}}, P)$ has the strong domination property. Let $t \in \mathcal{T}_n$ with $n \geq 2$ and $P(t) > 0$, and let $\mathcal{G} = (V, A_0, r)$ be a TSLP in normal form with $\mathsf{val}(\mathcal{G}) = t$. We have $H(G) \leq -\log_2 P(t) + O(|\mathcal{G}|) + O(|\mathcal{G}| \cdot \log_2(n/|\mathcal{G}|))$.*

Let us fix the grammar-based tree compressor $\psi : t \mapsto \mathcal{G}_t$ from Theorem 1; thus $\gamma_\psi(n) \in O(1/\log n)$. We define the tree encoder $E_{\text{tslp}} : \mathcal{T} \to \{0,1\}^*$ by $E_{\text{tslp}}(t) = B(\mathcal{G}_t)$.

Theorem 2. *If $\mathcal{S} = ((\mathcal{F}_i)_{i\in\mathbb{N}}, P)$ has the strong domination property, $n_i := \min\{|t| \mid t \in \mathcal{F}_i\}$ and $n_i < n_{i+1}$ for all $i \in \mathbb{N}$, then, $R(E_{tslp}, \mathcal{S}, i) \leq O(\log\log n_i / \log n_i)$.*

Note that the minimal size of a tree in $\mathcal{T}_{i+1}$ (resp. $\mathcal{T}^i$) is $i+1$. Hence, Thm. 2 yields:

Corollary 3. *If $\mathcal{S}$ is a leaf-centric or depth-centric tree source with the strong domination property, then $R(E_{tslp}, \mathcal{S}, i) \leq O(\log\log i / \log i)$.*

In the rest of the paper, we present two classes of tree sources having the strong domination property. Recall the definition of the class of mappings Σ_{leaf} (resp., Σ_{depth}) by (2) and (4) (resp., (3) and (5)) in Sec. II-B. A mapping $\sigma \in \Sigma_{\text{leaf}}$ (resp. $\sigma \in \Sigma_{\text{depth}}$) is monotone, if $\sigma(i,j) \geq \sigma(i, j+1)$ and $\sigma(i,j) \geq \sigma(i+1, j)$ for all $i, j \geq 1$ (resp., $i, j \geq 0$). The following theorem allows to apply Cor. 3 to the tree source $\mathcal{S}^l_\sigma$ (resp., $\mathcal{S}^d_\sigma$) for a monotone σ.

Theorem 3. *If $\sigma \in \Sigma_{leaf}$ (resp., $\sigma \in \Sigma_{depth}$) is monotone, then the leaf-centric tree source $\mathcal{S}^l_\sigma$ (resp., the depth-centric tree source $\mathcal{S}^d_\sigma$) has the strong domination property.*

Example 5. *Consider $\sigma \in \Sigma_{leaf}$ with $\sigma(i,j) = 1/(i+j)$. It is clearly monotone. Hence, $R(E_{tslp}, \mathcal{S}^l_\sigma, i) \leq O(\log\log i / \log i)$. The tree source $\mathcal{S}^l_\sigma$ is the famous* binary search tree model*; see [9] for an investigation in the context of information theory.*

Acknowledgement

This work is supported by the DFG project LO 748/10-1.

References

[1] Y. Choi and W. Szpankowski. Compression of graphical structures: Fundamental limits, algorithms, and experiments. *IEEE Trans. Inf. Theory*, 58(2):620–638, 2012.

[2] T. M. Cover. Enumerative source encoding. *IEEE Trans. Inf. Theory*, 19(1):73–77, 1973.

[3] T. M. Cover and J. A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006.

[4] D. Hucke and M. Lohrey. Universal tree source coding using grammar-based compression. *arXiv.org*, 2017. https://arxiv.org/abs/1701.08785.

[5] M. Ganardi, D. Hucke, A. Jeż, M. Lohrey, and E. Noeth. Constructing small tree grammars and small circuits for formulas. *J. Comput. Syst. Sci.*, 86:136–158, 2016.

[6] L. Hübschle-Schneider and R. Raman. Tree compression with top trees revisited. In *Proc. SEA 2015*, LNCS 9125, pages 15–27. Springer, 2015.

[7] J. C. Kieffer. A survey of Bratteli information source theory. In *Proc. ISIT 2016*, pages 16–20. IEEE, 2016.

[8] J, C. Kieffer and E. H. Yang. Grammar-based codes: A new class of universal lossless source codes. *IEEE Trans. Inf. Theory*, 46(3):737–754, 2000.

[9] J. C. Kieffer, E. H. Yang, and W. Szpankowski. Structural complexity of random binary trees. In *Proc. ISIT 2009*, pages 635–639. IEEE, 2009.

[10] M. Lohrey. Algorithmics on SLP-compressed strings: A survey. *Groups Complexity Cryptology*, 4(2):241–299, 2012.

[11] M. Lohrey. Grammar-based tree compression. In *Proc. DLT 2015*, LNCS 9168, pages 46–57. Springer, 2015.

[12] A. Magner, K. Turowski, and W. Szpankowski. Lossless compression of binary trees with correlated vertex names. In *Proc. ISIT 2016*, pages 1217–1221. IEEE, 2016.

[13] J. Zhang, E. H. Yang, and J. C. Kieffer. A universal grammar-based code for lossless compression of binary trees. *IEEE Trans. Inf. Theory*, 60(3):1373–1386, 2014.

[14] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Trans. Inf. Theory*, 24(5):530–536, 1977.