Universal Tree Source Coding Using Grammar-Based Compression

Moses Ganardi, Danny Hucke, Markus Lohrey, and Louisa Seelbach Benkner

***Abstract*—The problem of universal source coding for binary trees is considered. Zhang, Yang, and Kieffer derived upper bounds on the average-case redundancy of codes based on Directed Acyclic Graph (DAG) compression for binary tree sources with certain properties. In this paper, a natural class of binary tree sources is presented such that the demanded properties are fulfilled. Moreover, for both subclasses considered in the paper of Zhang, Yang, and Kieffer, their result is improved by deriving bounds on the maximal pointwise redundancy (or worst-case redundancy) instead of the average-case redundancy. Finally, using context-free tree grammars instead of DAGs, upper bounds on the maximal pointwise redundancy for certain binary tree sources are derived. This yields universal codes for new classes of binary tree sources.**

***Index Terms*—Grammar-based compression, minimal DAG representation, binary trees, universal source coding, lossless compression.**

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. In a series of papers, Cosman, Kieffer, Nelson, and Yang developed grammar-based codes that are universal for the class of finite state sources [16], [17], [18], [27]. 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 [20] 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 which compute from a given input string w of length n an SLP $\mathcal{G}_w$ of size $\mathcal{O}(n/\log n)$ (the size of an SLP is the total number of symbols in all right-hand sides of the grammar) [16]; the best known example is probably the LZ78 algorithm [29]. By combining any of these algorithms with the binary encoder B for SLPs from [16], one obtains a grammar-based encoder $E : \Sigma^* \to \{0,1\}^*$, whose worst-case redundancy for input strings of length n is bounded by $\mathcal{O}(\log\log n/\log n)$ for every finite state information source over the alphabet Σ. Here, the worst-case redundancy for strings of length n is defined as

$$\max_{w\in\Sigma^n, P(w)>0} \frac{1}{n} \cdot (|E(w)| + \log_2 P(w)),$$

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 [19], [23], [28] and graphs [4], [15]. In this paper, we are concerned with universal source coding for binary trees. Binary trees are ubiquitous in computer science. They appear in various efficient data structures (e.g., binary search trees, Cartesian trees, red-black trees, AVL-trees; see [3] for details). Large binary trees are also obtained when unranked trees (e.g., XML document trees) are encoded using the first-child next-sibling encoding. The resulting trees turned out to be highly compressible with so called tree straight-line programs [22]; more on them later. In their recent paper [28], Kieffer, Yang, and Zhang started to extend their work on grammar-based source coding from strings to binary trees. For this, they first represent the input tree t by its *minimal directed acyclic graph* $\mathcal{D}_t$ (the minimal DAG of t). This is the directed acyclic graph obtained by identifying multiple occurrences of the same subtree from t. In a second step, the minimal DAG $\mathcal{D}_t$ is encoded by a binary string $B(\mathcal{D}_t)$; this step is similar to the binary coding of SLPs from [16]. Combining both steps yields a tree encoder $E_{\text{dag}} : \mathcal{T} \to \{0,1\}^*$, where $\mathcal{T}$ denotes 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 [28] by the notion of a *structured tree source*. While the shortened term "tree source" is also used in the literature to describe a different concept in source coding (see e.g. [26]), we follow the definition in [28] where it describes 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 $\mathcal{T}$. The main cases considered in [28] as well as in this paper are:

- *leaf-centric sources*, where F_n is the set of all binary trees with n leaves, and
- *depth-centric sources*, where F_n is the set of all binary trees of depth n.

Then, the authors of [28] introduce two properties on binary tree sources:

Manuscript received November 15, 2017, revised October 19, 2018; accepted May 9, 2019. A short version of this paper appeared in the Proceedings of ISIT 2017 [14]. This work has been supported by the DFG research project LO 748/10-1 (QUANT-KOMP).

M. Ganardi, D. Hucke, M. Lohrey and L. Seelbach Benkner are with the Department of Electrical Engineering and Computer Science, University of Siegen, 57076 Siegen, Germany. E-mail: {ganardi,hucke,lohrey,seelbach}@eti.uni-siegen.de.

(i) the domination property (see Section III, where it is called the weak domination property) and
(ii) 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 technical main result of [28] states that for every structured tree source $(P_n)_{n\in\mathbb{N}}$ satisfying the domination property and the representation ratio negligibility property the *average-case redundancy*

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

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 [28]. One is a class of leaf-centric sources, the other one is a class of depth-centric sources. Both classes have the property that every tree with a non-zero probability is balanced in a certain sense, the precise definitions can be found in Section III-C and Section III-D. As a first contribution, we show that for these sources not only the average-case redundancy but also the worst-case redundancy

$$\max_{t\in F_n, P_n(t)>0} \frac{1}{|t|}\cdot(|E_{\text{dag}}(t)|+\log_2 P_n(t)) \qquad (2)$$

converges to zero for $n\to\infty$. More precisely, we show that (2) is bounded by $\mathcal{O}(\log\log n/\log n)$ (respectively, $\mathcal{O}((\log\log n)^2/\log n)$) for the presented class of leaf-centric tree sources (respectively, depth-centric tree sources). To prove this, we use results from [11], [13] according to which the size of the minimal DAG of a suitably balanced binary tree of size n is bounded by $\mathcal{O}(n/\log n)$, respectively $\mathcal{O}(n\cdot\log\log n/\log n)$.

As a second main contribution we introduce a new class of leaf-centric tree sources having the domination property and the representation ratio negligibility property. More precisely, the average compression ratio achieved by the minimal DAG is bounded by $\mathcal{O}(1/\log n)$. Intuitively, these leaf-centric tree sources produce balanced trees with higher probabilities than unbalanced ones, but nevertheless (and in contrast to the two specific classes of tree sources studied in [28]) may produce every tree (also unbalanced ones) with non-zero probability. Examples for such tree sources are the binary search tree model and the binomial random tree model, see [19] and Example 5. Together with the results from [28] we obtain the upper bound of $\mathcal{O}(\log\log n/\log n)$ for the average-case redundancy (1) for these tree sources.

Our third 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 Section II-C for the precise definition and [21] 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 [11], 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 $\mathcal{O}(n/\log n)$. 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 Section IV-B we define a binary encoding B of TSLPs similar to the ones for SLPs [16] and DAGs [28]. 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 (that is defined as in (2) with E_{dag} replaced by E_{tslp}) is bounded by $\mathcal{O}(\log\log n/\log n)$ for every structured tree source that satisfies the *strong domination property* defined in Section IV-C. The strong domination property is a strengthening of the domination property from [28], and this is what we have to pay extra for our TSLP-based encoding in contrast to the DAG-based encoding from [28]. On the other hand, our approach has two main advantages over [28]:

- The representation ratio negligibility property from [28] is no longer needed.
- 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 [11] computes a TSLP of worst-case size $\mathcal{O}(n/\log n)$ for a binary tree of size n.

Finally, we present a class of leaf-centric sources (Section III-C) as well as a class of depth-centric sources (Section III-D) having the strong domination property. These classes are orthogonal to the classes considered in [28].

II. Preliminaries

In this section, we introduce some basic definitions concerning information theory (Section II-A), binary trees (Section II-B) and tree straight-line programs (Section II-C). The latter are our key formalism for the compression of binary trees.

With $\mathbb{N}$ we denote the natural numbers including 0. We use the standard $\mathcal{O}$-notation and if b is a constant, then we just write $\mathcal{O}(\log n)$ for $\mathcal{O}(\log_b n)$. For the unit interval $\{r\in\mathbb{R}\mid 0\le r\le 1\}$ we write $[0,1]$.

A. Empirical distributions and empirical entropy

Let $\overline{a}=(a_1,a_2,\ldots,a_n)$ be a tuple of elements that are from some (not necessarily finite) set A. The *empirical distribution* $p_{\overline{a}}:\{a_1,a_2,\ldots,a_n\}\to\mathbb{R}$ of $\overline{a}$ is defined by

$$p_{\overline{a}}(a)=\frac{|\{i\mid 1\le i\le n,\ a_i=a\}|}{n}.$$

We use this definition also for words over some alphabet by identifying a word $w=a_1a_2\cdots a_n$ with the tuple $(a_1,a_2,\ldots,a_n)$. The *unnormalized empirical entropy* of $\overline{a}$ is

$$H(\overline{a})=-\sum_{i=1}^{n}\log_2 p_{\overline{a}}(a_i).$$

A well-known generalization of Shannon's inequality states that for all real numbers $p_1,\ldots,p_k,q_1,\ldots,q_k>0$, if $\sum_{i=1}^k p_i=1\ge\sum_{i=1}^k q_i$ then

$$\sum_{i=1}^{k}-p_i\log_2(p_i)\le\sum_{i=1}^{k}-p_i\log_2(q_i);$$

see [1] for a proof. As a consequence, for a tuple $\overline{a} = (a_1, a_2, \ldots, a_n)$ with $a_1, \ldots, a_n \in A$ and real numbers $q(a) > 0$ ($a \in A$) with $\sum_{i=1}^{n} q(a_i) \leq 1$ we have

$$\sum_{i=1}^{n} -\log_2(p_{\overline{a}}(a_i)) \leq \sum_{i=1}^{n} -\log_2(q(a_i)). \tag{3}$$

B. Trees, tree sources, and tree compressors

In the literature one finds several different definitions of binary trees. In this paper, a binary tree is a finite rooted tree, where every node is either a leaf or has a left and a right child. With $\mathcal{T}$ we denote the set of all such binary trees. We identify $\mathcal{T}$ with the set of terms that are built from the binary symbol f and the constant a. That means that an internal node of a binary tree is labelled with the symbol f and has exactly two children whereas a leaf node is labelled with the symbol a and has zero children. Formally, $\mathcal{T}$ is the smallest set of terms such that (i) $a \in \mathcal{T}$ and (ii) if $t_1, t_2 \in \mathcal{T}$ then also $f(t_1, t_2) \in \mathcal{T}$. For example, the term $f(a, a)$ is the tree which has a root node labelled with f and the two children of the root are both leaves labelled with a. Another example is the tree $f(f(f(a, a), a), f(a, a))$, which is depicted on the left of Figure 2. With $|t|$ we denote the number of leaves of t, i.e. the number of occurrences of the constant a in t. Let $\mathcal{T}_n = \{t \in \mathcal{T} \mid |t| = n\}$ for $n \geq 1$. We have $|\mathcal{T}_n| = C_{n-1}$, where C_k is the k^{th} Catalan number. These numbers satisfy the following well-known asymptotic estimate:

$$C_k \sim \frac{4^k}{\sqrt{\pi} k^{\frac{3}{2}}},$$

see e.g. [9]. In fact, we have $C_k \leq 4^k$ for all $k \geq 0$. 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. The *leaf-size* of a node v of t is the size of the subtree of t with root node v.

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. Formally, the set of contexts $\mathcal{C}$ is the smallest set such that (i) $x \in \mathcal{C}$ and (ii) if $s \in \mathcal{C}$ and $t \in \mathcal{T}$ then also $f(s, t), f(t, s) \in \mathcal{C}$. For example the term $f(a, x)$ is the context which has a root node labelled with f, the left child of the root is a leaf labelled with a and the right child is the single leaf labelled with the parameter x. 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 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. This can be inductively defined by the following rules, where $s' \in \mathcal{C}$ and $t' \in \mathcal{T}$: $x[t] = t$, $f(s', t')[t] = f(s'[t], t')$, and $f(t', s')[t] = f(t', s'[t])$. For example $s = f(a, x)$ and $t = f(a, a)$ yields $s[t] = f(a, f(a, a))$. By induction on the size of the context s it follows directly that $|s[t]| = |s| + |t|$. Here, it is important that the unique occurrence of the parameter x does not count to the size of the context. The depth $d(t)$ of a context $t \in \mathcal{C}$ is defined as the depth of the tree $t[a]$. We say that a context s occurs in a tree t if there is a tree u such that $s[u]$ is a subtree of t. For instance, the context $f(a, x)$ occurs in $f(a, f(a, a))$, and there are two such occurrences (since $f(a, x)[f(a, a)] = f(a, f(a, a))$ and $f(a, x)[a] = f(a, a)$ are both subtrees of $f(a, f(a, a))$).

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

- $\mathcal{F}_i \subseteq \mathcal{T}$ is non-empty and finite for every $i \geq 0$,
- $\mathcal{F}_i \cap \mathcal{F}_j = \emptyset$ for $i \neq j$ and $\bigcup_{i \geq 0} \mathcal{F}_i = \mathcal{T}$, 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 $i \geq 0$, 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*).

A *tree encoder* is an injective mapping $E : \mathcal{T} \to \{0, 1\}^*$ such that the 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* of E with respect to the fixed 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

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

(the maximum is taken over all trees $t \in \mathcal{F}_i$ such that $P(t) > 0$). The worst-case redundancy is also known as the *maximal pointwise redundancy*. Moreover, we define the *average-case redundancy* of E with respect to the tree source $\mathcal{S} = ((\mathcal{F}_i)_{i \in \mathbb{N}}, P)$ as the mapping $i \mapsto R_{\varnothing}(E, \mathcal{S}, i)$ ($i \in \mathbb{N}$) with

$$R_{\varnothing}(E, \mathcal{S}, i) = \sum_{t \in \mathcal{F}_i, P(t) > 0} \frac{1}{|t|} \cdot (|E(t)| + \log_2 P(t)) \cdot P(t).$$

C. Tree straight-line programs

We now introduce tree straight-line programs for binary trees. Let $V = V_0 \cup V_1$ be a finite alphabet with $V_0 \cap V_1 = \emptyset$. Elements of V are called *nonterminals*, and elements of V_0 (resp., V_1) are called nonterminals of rank zero (resp., rank one). We assume that V_0 is not empty and that V is disjoint from the set $\{f, a, x\}$, which are the labels used for binary trees and contexts. We will consider trees, where every node is labelled with a symbol from $\{f, a\} \cup V$. Leaves have to be labelled with symbols from $V_0 \cup \{a\}$ and internal nodes have to be labelled with symbols from $V_1 \cup \{f\}$. An internal node that is labelled with f has exactly two children (as for trees from $\mathcal{T}$), and an internal node that is labelled with a symbol from V_1 has exactly one child. Such trees can be conveniently described by terms, as we did for trees from $\mathcal{T}$. Formally, we define the set of terms $\mathcal{T}_V$ as the smallest set such that

(i) $a \in \mathcal{T}_V$,
(ii) if $A \in V_0$ then $A \in \mathcal{T}_V$,
(iii) if $A \in V_1$ and $t \in \mathcal{T}_V$ then $A(t) \in \mathcal{T}_V$, and

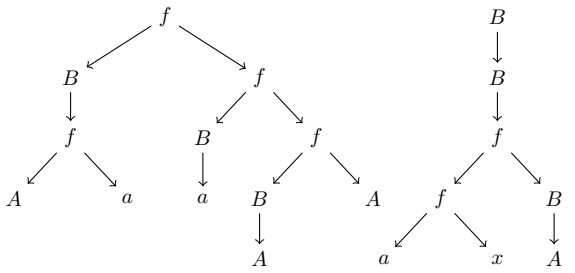

Fig. 1. A tree from $\mathcal{T}_V$ (left) and a context from $\mathcal{C}_V$ (right), where $V_0 = \{A\}$, $V_1 = \{B\}$ and $V = V_0 \cup V_1$.

(iv) if $t_1, t_2 \in \mathcal{T}_V$ then $f(t_1, t_2) \in \mathcal{T}_V$.

Note that $A(t)$ denotes the tree whose root node is labelled with A and the subtree rooted in the unique child of the root is t.

With $\mathcal{C}_V$ we denote the corresponding set of all contexts, i.e., the set of trees obtained from a tree in $\mathcal{T}_V$ by relabelling exactly one leaf node by the parameter symbol x. Formally, we can define $\mathcal{C}_V$ as the smallest set of terms such that

(i) $x \in \mathcal{C}_V$,
(ii) if $A \in V_1$ and $t \in \mathcal{C}_V$ then $A(t) \in \mathcal{C}_V$, and
(iii) if $s \in \mathcal{C}_V$ and $t \in \mathcal{T}_V$ then $f(s,t), f(t,s) \in \mathcal{C}_V$.

Note that $\mathcal{T} \subseteq \mathcal{T}_V$ and $\mathcal{C} \subseteq \mathcal{C}_V$.

Example 1. *Let $V_0 = \{A\}$, $V_1 = \{B\}$ and $V = V_0 \cup V_1$. An example for a tree $t \in \mathcal{T}_V$ is*

$$t = f(B(f(A,a)), f(B(a), f(B(A), A))).$$

This tree is depicted on the left of Figure 1. An example for a context $c \in \mathcal{C}_V$ is $c = B(B(f(f(a,x), B(A))))$. This context is depicted on the right of Figure 1.

A *tree straight-line program $\mathcal{G}$, TSLP* for short, is a tuple (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 a function which assigns to each nonterminal its unique right-hand side. It is required that if $A \in V_0$ (respectively, $A \in V_1$), then $r(A) \in \mathcal{T}_V$ (respectively, $r(A) \in \mathcal{C}_V$). Furthermore, the binary relation

$$\{(A, B) \in V \times V \mid B \text{ is a label in } r(A)\}$$

needs to be acyclic. This allows to define for every $t \in \mathcal{T}_V$ a tree $\mathsf{val}_\mathcal{G}(t) \in \mathcal{T}$ and for every $t \in \mathcal{C}_V$ a context $\mathsf{val}_\mathcal{G}(t) \in \mathcal{C}$ inductively by the following rules ("val" stands for "evaluation"):

- $\mathsf{val}_\mathcal{G}(a) = a$ 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 $t_1, t_2 \in \mathcal{T}_V \cup \mathcal{C}_V$ (and $t_1 \in \mathcal{T}_V$ or $t_2 \in \mathcal{T}_V$ since there is at most one parameter in $f(t_1,t_2)$),
- $\mathsf{val}_\mathcal{G}(A) = \mathsf{val}_\mathcal{G}(r(A))$ for $A \in V_0$,
- $\mathsf{val}_\mathcal{G}(A(s)) = \mathsf{val}_\mathcal{G}(r(A))[\mathsf{val}_\mathcal{G}(s)]$ for $A \in V_1$, $s \in \mathcal{T}_V \cup \mathcal{C}_V$ (note that $\mathsf{val}_\mathcal{G}(r(A))$ is a context t', so we can build $t'[\mathsf{val}_\mathcal{G}(s)]$).

The intuition behind this definition is as follows: To define $\mathsf{val}_\mathcal{G}(t)$ for $t \in \mathcal{T}_V$ or $t \in \mathcal{C}_V$ one starts with the term t and applies the rewrite rules $A \to r(A)$ for $A \in V$ as long as possible until all nonterminals are eliminated. Readers that are familiar with the notion of context-free tree grammars will notice that a TSLP is a context-free tree grammar that produces a unique tree.

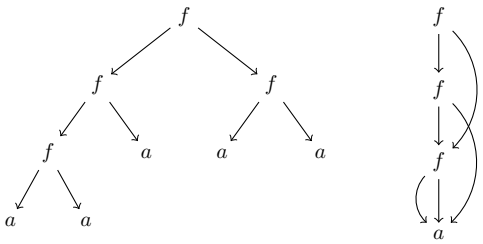

Fig. 2. A tree (left) and its minimal DAG (right).

The tree defined by $\mathcal{G}$ is $\mathsf{val}(\mathcal{G}) = \mathsf{val}_\mathcal{G}(A_0) \in \mathcal{T}$ (note that $A_0 \in V_0$). Moreover, for $A \in V_1$ we also write $\mathsf{val}_\mathcal{G}(A)$ for the context $\mathsf{val}_\mathcal{G}(A(x))$ (note that A itself is neither a context nor a tree by our definition).

Example 2. *Let $\mathcal{G} = (\{A_0, A_1, A_2\}, A_0, r)$ be a TSLP with $A_0, A_1 \in V_0, A_2 \in V_1$ and*

$$\begin{aligned} r(A_0) &= f(A_1, A_2(a)), \\ r(A_1) &= A_2(A_2(a)), \\ r(A_2) &= f(x, a). \end{aligned}$$

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 the introduction it is mentioned that an advantage of TSLPs is the ability to share repeated occurrences of a context in a tree. In the above example one can see that the context $f(x,a)$ occurs three times in $\mathsf{val}(\mathcal{G}) = f(f(f(a,a),a), f(a,a))$ (starting at each internal node except the root node) and we use the nonterminal A_2 to represent those occurrences in the TSLP.

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 Section IV-A.

III. Tree compression with DAGs

In this section we sharpen some of the results from [28], where universal source coding of binary trees using minimal DAGs (directed acyclic graphs) is investigated. In [28], only bounds on the average-case 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 commonly used compact tree compression scheme is obtained by writing down repeated subtrees only once. In that case all occurrences except for the first are replaced by a pointer to the first one. This leads to a node-labelled (minimal) DAG for a tree. In Figure 2 the minimal DAG of the tree considered in Example 2 is depicted. A DAG can be seen as a TSLP where only nonterminals of rank 0 occur. In this case, each node of the DAG (except the single a-labelled node) is

represented by a nonterminal and the edges are represented by the right-hand sides of the rules (see Example 3 for the TSLP corresponding to Figure 2).

Formally, 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 $\alpha_1, \alpha_2 \in \{a, A_{i+1}, \ldots, A_{n-1}\}$. 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 $\mathcal{G}_a = (\{A_0\}, A_0, A_0 \mapsto a)$. We define the size of a DAG as $|\mathcal{D}| = n + 1$.

Example 3. *Consider the tree*

$$t = f(f(f(a, a), a), f(a, a))$$

depicted in Figure 2. The minimal DAG of t is

$$\mathcal{D} = (\{A_0, A_1, A_2\}, A_0, r)$$

with $r(A_0) = f(A_1, A_2)$, $r(A_1) = f(A_2, a)$ and $r(A_2) = f(a, a)$.

In contrast to general TSLPs, every binary tree t has a unique (up to renaming of nonterminals) minimal DAG $\mathcal{D}_t$ with $\mathsf{val}(\mathcal{D}_t) = t$. The size of $\mathcal{D}_t$, denoted with $|\mathcal{D}_t|$, 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 $\alpha_j = a$ if $t_j = a$ and α_j is the nonterminal corresponding to the subtree t_j if $|t_j| \geq 2$ ($j \in \{1, 2\}$). We will only use this minimal DAG $\mathcal{D}_t$ in the sequel.

Example 4. *Consider the tree*

$$t_n = f(f(f(\cdots f(a, a), \cdots a), a), a),$$

where f occurs n times. We have $|t_n| = n + 1$. The minimal DAG of t_n is $(\{A_0, A_1, \ldots, A_{n-1}\}, A_0, r_n)$, 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 $n + 1$.

The above example shows that in the worst-case, the size of the minimal DAG is not smaller than the size of the tree.

B. Universal source coding with DAGs

The following condition on a tree source was introduced in [28], where it is called the domination property (later, we will introduce a strong domination property): Let $((\mathcal{F}_i)_{i\in\mathbb{N}}, P)$ be a tree source as defined in Section II-B. We say that $((\mathcal{F}_i)_{i\in\mathbb{N}}, P)$ has the *weak domination property* if there exists a mapping $\lambda : \mathcal{T} \to \mathbb{R}_{>0}$ with the following properties:

(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) 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 [28], 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 [28, Theorem 2], where $E_{\text{dag}} : \mathcal{T} \to \{0, 1\}^*$ is the tree encoder with $E_{\text{dag}}(t) = B(\mathcal{D}_t)$.

Lemma 1. *Assume that $((\mathcal{F}_i)_{i\in\mathbb{N}}, P)$ has the weak domination property. Let $t \in \mathcal{T}_n$ with $n \geq 2$ and $P(t) > 0$, and let $\mathcal{D}_t$ be the minimal DAG for t. We have*

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

The following bound on the average-case redundancy was derived in [28, Theorem 2] from Lemma 1:

Theorem 1. *Consider the mapping $g(x) = x \cdot \log_2(1/x)$ and assume that the tree source $\mathcal{S} = ((\mathcal{F}_i)_{i\in\mathbb{N}}, P)$ has the weak domination property. There exists a positive real number C, depending only on the tree source, such that for all $i \in \mathbb{N}$ we have*

$$R_{\varnothing}(E_{dag}, \mathcal{S}, i) \leq C \cdot g\Big(\sum_{t\in\mathcal{F}_i} P(t) \cdot \frac{|\mathcal{D}_t|}{|t|}\Big).$$

This bound is used in [28] to show that for certain leaf-centric and depth-centric tree sources the encoding E_{dag} is universal in the sense that the average-case redundancy converges to zero. Here, we want to show that for the same tree sources already the worst-case redundancy converges to zero. Let us first define the specific classes of tree sources studied in [28].

C. Leaf-centric binary tree sources

We recall the definition of a natural class of leaf-centric tree sources from [28]: Let Σ_{leaf} be the set of all functions $\sigma : (\mathbb{N} \setminus \{0\}) \times (\mathbb{N} \setminus \{0\}) \to [0, 1]$ such that for all $n \geq 2$:

$$\sum_{i=1}^{n-1} \sigma(i, n - i) = 1. \tag{4}$$

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

$$P_\sigma(a) = 1, \tag{5}$$
$$P_\sigma(f(s, t)) = \sigma(|s|, |t|) \cdot P_\sigma(s) \cdot P_\sigma(t). \tag{6}$$

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

Example 5. *Here are three examples of leaf-centric tree sources of the above form, which are also discussed in [19] with respect to their entropy rates:*

- *The* binary search tree model $((\mathcal{T}_i)_{i\geq 1}, P_{\sigma_{bst}})$*, where*

$$\sigma_{bst}(k, n - k) = 1/(n - 1)$$

for all $1 \leq k \leq n - 1$.

- *The* uniform model $((\mathcal{T}_i)_{i\geq 1}, P_{\sigma_{uni}})$*, where*

$$\sigma_{uni}(k, n - k) = \frac{C_{k-1} \cdot C_{n-k-1}}{C_{n-1}}$$

for all $1 \leq k \leq n - 1$. Here, $P_{\sigma_{uni}}$ yields the uniform distribution on every $\mathcal{T}_i$.

- *The* binomial random tree model $((\mathcal{T}_i)_{i\geq 1}, P_{\sigma_{bin}})$ *where*

$$\sigma_{bin}(k, n-k) = \binom{n-2}{k-1} \cdot \frac{1}{2^{n-2}}$$

for all $1 \leq k \leq n-1$.

The following result is implicitly shown in [28] (see also the proof of Theorem 5).

Lemma 2. *For every* $\sigma \in \Sigma_{leaf}$*, the leaf-centric tree source* $((\mathcal{T}_i)_{i\geq 1}, P_\sigma)$ *has the weak domination property.*

1) Worst-case redundancy: We say that a mapping $\sigma \in \Sigma_{\text{leaf}}$ is *leaf-balanced* if there exists a constant c such that for all $(i,j) \in (\mathbb{N}\setminus\{0\}) \times (\mathbb{N}\setminus\{0\})$ with $\sigma(i,j) > 0$ we have

$$\frac{i+j}{\min\{i,j\}} \leq c.$$

In [28] it is shown that for a leaf-balanced $\sigma \in \Sigma_{\text{leaf}}$, the tree source $((\mathcal{T}_i)_{i\geq 1}, P_\sigma)$ has the so called representation ratio negligibility property. This means 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 [11], we show the following stronger property.

Lemma 3. *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* $|\mathcal{D}_t| \leq \alpha \cdot n/\log_2 n$.

Proof. In [11] the authors introduce the notion of β-balanced trees. Let v be a non-leaf node in $t \in \mathcal{T}$ and s be the subtree of t with root node v. Since v is a non-leaf node we have $s = f(t_1, t_2)$ for binary trees t_1 and t_2. Then v is *β-balanced* if $|t_1| \leq \beta\cdot|t_2|$ and $|t_2| \leq \beta\cdot|t_1|$. The tree t is β-balanced if for all non-leaf nodes u and v in t such that u is a child node of v, at least one of the nodes u, v is β-balanced. It is shown in [11] that for every constant β there exists a constant α (depending only on β) such that the minimal DAG $\mathcal{D}_t$ of a β-balanced tree $t \in \mathcal{T}_n$ has size at most $\alpha \cdot n/\log_2 n$. It follows that we only need to show that a tree $t \in \mathcal{T}_n$ with $P_\sigma(t) > 0$ is β-balanced for a constant β. Since the mapping σ is leaf-balanced, every subtree $f(t_1, t_2)$ of t satisfies $|t_1| + |t_2| \leq c \cdot \min\{|t_1|, |t_2|\}$, where $c \geq 1$ is a constant. Without loss of generality assume that $|t_1| \leq |t_2|$. We get $|t_1| \leq c \cdot |t_2|$ and

$$|t_2| \leq |t_1| + |t_2| \leq c \cdot \min\{|t_1|, |t_2|\} = c \cdot |t_1|,$$

which shows that t is c-balanced. □

Corollary 1. *Let* $\sigma \in \Sigma_{leaf}$ *be leaf-balanced, and let* $\mathcal{S} = ((\mathcal{T}_i)_{i\geq 1}, P_\sigma)$ *be the corresponding leaf-centric tree source. Then, we have* $R(E_{dag}, \mathcal{S}, i) \leq \mathcal{O}(\log\log i/\log i)$.

Proof. Let α be the constant from Lemma 3. Let $t \in \mathcal{T}_n$ such that $P_\sigma(t) > 0$. Lemma 3 implies $|\mathcal{D}_t| \leq \alpha \cdot n/\log_2 n$. With Lemma 1 and 2 we get

$$\begin{aligned}
&\frac{1}{n} \cdot (|E_{\text{dag}}(t)| + \log_2(P_\sigma(t))) \\
&\leq \mathcal{O}(|\mathcal{D}_t|/n) + \mathcal{O}(|\mathcal{D}_t|/n \cdot \log_2(n/|\mathcal{D}_t|)) \\
&\leq \mathcal{O}(\log\log n/\log n)
\end{aligned}$$

This proves the corollary. □

2) Average-case redundancy: In this subsection we present another class of leaf-centric binary tree sources $\Sigma^{\text{m}}_{\text{leaf}} \subseteq \Sigma_{\text{leaf}}$ such that for every $\sigma \in \Sigma^{\text{m}}_{\text{leaf}}$ the average-case redundancy $R_{\varnothing}(E_{\text{dag}}, \mathcal{S}, i)$ of the corresponding tree source $\mathcal{S} = ((\mathcal{T}_i)_{i\geq 1}, P_\sigma)$ is bounded by $\mathcal{O}(\log\log i/\log i)$ and hence converges to zero.

For $\sigma \in \Sigma_{\text{leaf}}$ we define the mapping $\sigma^* : (\mathbb{N}\setminus\{0\}) \times (\mathbb{N}\setminus\{0\}) \to [0,1]$ by

$$\sigma^*(i,j) = \frac{1}{2}(\sigma(i,j) + \sigma(j,i)).$$

Note that $\sigma^*(i,j) = \sigma^*(j,i)$ and $\sum_{i=1}^{n-1}\sigma^*(i,n-i) = 1$. This implies

$$\sum_{i=k}^{n-1}\sigma^*(i,n-i) \leq \frac{1}{2} \tag{7}$$

for $k > n/2$.

Let $\Sigma^{\text{m}}_{\text{leaf}} \subseteq \Sigma_{\text{leaf}}$ denote the set of mappings σ which satisfy

$$\sigma^*(k,n-k) \leq \sigma^*(k+1,n-k-1)$$

for all integers $n \geq 1$ and $1 \leq k < \lfloor\frac{n}{2}\rfloor$. Since the mapping σ^* is symmetric in its first and second argument, this is equivalent to

$$\sigma^*(k,n-k) \geq \sigma^*(k+1,n-k-1)$$

for all integers $n \geq 1$ and $\lceil\frac{n}{2}\rceil \leq k \leq n-2$. For the set of leaf-centric binary tree sources induced by a mapping $\sigma \in \Sigma^{\text{m}}_{\text{leaf}}$ we find the following:

Theorem 2. *Let* $\sigma \in \Sigma^{m}_{leaf}$*. The average size of the minimal DAG of a binary tree* $t \in \mathcal{T}_n$ *with respect to the probability mass function* P_σ *on* $\mathcal{T}_n$ *satisfies*

$$\sum_{t\in\mathcal{T}_n} P_\sigma(t)\cdot|\mathcal{D}_t| \leq \mathcal{O}\left(\frac{n}{\log n}\right). \tag{8}$$

In the terms and definitions of [28], Theorem 2 states that the class of leaf-centric binary tree sources corresponding to a mapping $\sigma \in \Sigma^{\text{m}}_{\text{leaf}}$ satisfies the representation ratio neglibility property.

Example 6. *Recall the three leaf-centric binary tree sources from Example 5. The mappings* σ_{bst} *and* σ_{bin} *belong to* Σ^{m}_{leaf}*. Hence, for these sources the average size of the minimal DAG is bounded by* $\mathcal{O}(n/\log n)$*. For the binary search tree model, this bound was already shown in [8]; and a matching lower bound was shown in [7]. In contrast,* σ_{uni} *does not belong to* Σ^{m}_{leaf}*. In fact, in [10] (see also [2], [25]) it is shown that under the uniform distribution on* $\mathcal{T}_n$*, the average size of the minimal DAG is* $\Theta(n/\sqrt{\log n})$.

Our proof of Theorem 2 uses the cut-point argument which is also applied in [8]: The size of the minimal DAG of a tree t is bounded by (i) the number of nodes of t that have leaf size larger than a carefully chosen cut-point b and (ii) the number of *all* binary trees with at most b leaves (irrespective of their (non)occurrence in t).

Let us start with some formal definitions. For a binary tree $t \in \mathcal{T}$ and an integer $k \geq 1$ let $N(t,k)$ denote the number of nodes of t of leaf-size greater than k. Moreover, for a mapping

$\sigma \in \Sigma_{\text{leaf}}$ and integers $b \geq 1$ and $n \geq 1$ let $E_{\sigma,b}(n)$ denote the expected value of $N(t,b)$ with respect to the probability mass function P_σ on the set of binary trees $\mathcal{T}_n$:

$$E_{\sigma,b}(n) = \sum_{t \in \mathcal{T}_n} P_\sigma(t) \cdot N(t,b).$$

We find $E_{\sigma,b}(n) = 0$ if $n \leq b$. Let $t = f(u,v) \in \mathcal{T}_n$ and let $b < n$. The number of nodes of t of leaf-size greater than b is composed of the number of nodes of the left subtree u of leaf-size greater than b plus the number of nodes of the right subtree v of leaf-size greater than b plus one (for the root):

$$N(t,b) = N(u,b) + N(v,b) + 1.$$

This yields the following recurrence relation for the expected value $E_{\sigma,b}(n)$:

$$E_{\sigma,b}(n) = 1 + \sum_{k=b+1}^{n-1} (\sigma(k,n-k) + \sigma(n-k,k)) \cdot E_{\sigma,b}(k). \tag{9}$$

Moreover, we find the following upper bound for $E_{\sigma,b}(n)$:

Lemma 4. *For a mapping $\sigma \in \Sigma^m_{leaf}$ and integers $n > b \geq 1$, we have*

$$E_{\sigma,b}(n) \leq \frac{2(n-1)}{b} - 1.$$

In the proof we use Chebyshev's sum inequality, see e.g. [12, Section 2.7]:

Lemma 5. *If $a_1 \leq a_2 \leq \cdots \leq a_n$ and $b_1 \geq b_2 \geq \cdots \geq b_n$ then*

$$\sum_{i=1}^{n} a_i b_i \leq \frac{1}{n} \cdot \left(\sum_{i=1}^{n} a_i \right) \cdot \left(\sum_{i=1}^{n} b_i \right).$$

Proof of Lemma 4. We prove the statement by induction on $n \geq b+1$. For the base case, let $n = b+1$. A binary tree $t \in \mathcal{T}_{b+1}$ has exactly one node of leaf-size greater than b, which is the root of t. Thus, $E_{\sigma,b}(b+1) = 1$.

For the induction step, assume that $n > b+1$ is such that $E_{\sigma,b}(k) \leq \frac{2(k-1)}{b} - 1$ for every integer $b+1 \leq k \leq n-1$. We distinguish two cases:

Case 1: $n < 2(b+1)$ and hence $\lceil \frac{n+1}{2} \rceil \leq b+1$. By equation (9), we have

$$E_{\sigma,b}(n) = 1 + 2 \sum_{k=b+1}^{n-1} \sigma^*(k,n-k) \cdot E_{\sigma,b}(k).$$

For $\lceil \frac{n+1}{2} \rceil \leq k \leq n-1$ we define $S(k)$ as

$$S(k) = \begin{cases} \frac{2(k-1)}{b} - 1 & \text{if } \ k \geq b+1 \\ 0 & \text{otherwise.} \end{cases}$$

By the induction hypothesis, we have $E_{\sigma,b}(k) \leq S(k)$ for every integer $b+1 \leq k \leq n-1$. Thus, as σ^* is non-negative, we find

$$E_{\sigma,b}(n) \leq 1 + 2 \sum_{k=\lceil \frac{n+1}{2} \rceil}^{n-1} \sigma^*(k,n-k) \cdot S(k).$$

As $\sigma \in \Sigma^{\text{m}}_{\text{leaf}}$, we find that $\sigma^*(k,n-k) \geq \sigma^*(k+1,n-k-1)$ for $\lceil \frac{n+1}{2} \rceil \leq k \leq n-2$. Moreover, $S(k)$ is monotonically increasing in k. We now obtain the following calculation where the first inequality follows from Chebyshev's sum inequality (Lemma 5) and the second inequality follows from (7).

$$\begin{aligned} E_{\sigma,b}(n) &\leq 1 + \frac{2}{\lfloor \frac{n-1}{2} \rfloor} \sum_{k=\lceil \frac{n+1}{2} \rceil}^{n-1} \sigma^*(k,n-k) \sum_{k=\lceil \frac{n+1}{2} \rceil}^{n-1} S(k) \\ &\leq 1 + \frac{2}{n-2} \sum_{k=\lceil \frac{n+1}{2} \rceil}^{n-1} S(k) \\ &= 1 + \frac{2}{n-2} \sum_{k=b+1}^{n-1} \left(\frac{2(k-1)}{b} - 1 \right) \\ &= \frac{2(n-1)}{b} - 1. \end{aligned}$$

The last equality follows by a straightforward computation.

Case 2: $n \geq 2(b+1)$. By rearranging the sum, we obtain from equation (9):

$$\begin{aligned} E_{\sigma,b}(n) &= 1 + \sum_{k=b+1}^{n-b-1} \sigma(k,n-k) \left(E_{\sigma,b}(k) + E_{\sigma,b}(n-k) \right) \\ &\quad + 2 \sum_{k=n-b}^{n-1} \sigma^*(k,n-k) E_{\sigma,b}(k). \end{aligned}$$

The induction hypothesis yields

$$\begin{aligned} E_{\sigma,b}(n) &\leq 1 + \sum_{k=b+1}^{n-b-1} \sigma(k,n-k) \left(\frac{2(n-2)}{b} - 2 \right) \\ &\quad + 2 \sum_{k=n-b}^{n-1} \sigma^*(k,n-k) \left(\frac{2(k-1)}{b} - 1 \right). \end{aligned}$$

We set $\alpha := \sum_{k=b+1}^{n-b-1} \sigma(k,n-k)$ with $0 \leq \alpha \leq 1$. In case $\alpha = 1$, the statement follows immediately. If $\alpha < 1$, we get

$$\begin{aligned} E_{\sigma,b}(n) &\leq 1 + \alpha \left(\frac{2(n-2)}{b} - 2 \right) + \\ &\quad 2(1-\alpha) \sum_{k=n-b}^{n-1} \frac{\sigma^*(k,n-k)}{1-\alpha} \left(\frac{2(k-1)}{b} - 1 \right). \end{aligned}$$

As $\sigma \in \Sigma^{\text{m}}_{\text{leaf}}$, we find that $\sigma^*(k,n-k) \geq \sigma^*(k+1,n-k-1)$ for $n-b \leq k \leq n-2$ (note that $n-b \geq \lceil \frac{n}{2} \rceil$). Applying Chebyshev's sum inequality, we obtain

$$\begin{aligned} E_{\sigma,b}(n) &\leq 1 + \alpha \left(\frac{2(n-2)}{b} - 2 \right) \\ &\quad + \frac{(1-\alpha)}{b} \sum_{k=n-b}^{n-1} \left(\frac{2(k-1)}{b} - 1 \right) \\ &= \frac{2n-\alpha-3}{b} - 1 \\ &\leq \frac{2n-2}{b} - 1, \end{aligned}$$

as $\alpha \geq 0$. This finishes the proof. □

With Lemma 4, we can prove Theorem 2 using the standard cut-point argument:

Proof of Theorem 2. Let $t \in \mathcal{T}_n$, with $n \geq 2$, and let $1 \leq b < n$. The number of different subtrees of t can be upper bounded by the sum of

(i) the number $N(t,b)$ of nodes of leaf-size at least $b+1$ and
(ii) the number of different subtrees of t with at most b leaves.

The number (ii) is bounded by the number of all binary trees with at most b leaves, which is $\sum_{k=0}^{b-1} C_k$, where C_k is the k^{th} Catalan number. Thus, we have

$$\begin{aligned} |\mathcal{D}_t| &\leq \sum_{k=0}^{b-1} C_k + N(t,b) \\ &\leq \sum_{k=0}^{b-1} 4^k + N(t,b) \\ &\leq \frac{1}{3}4^b + N(t,b). \end{aligned}$$

Let $b = \frac{1}{2}\lfloor \log_4(n) \rfloor$. We get $|\mathcal{D}_t| \leq \mathcal{O}\left(\frac{n}{\log n}\right) + N(t,b)$. Furthermore, with Lemma 4 we have

$$\begin{aligned} E_{\sigma,b}(n) &= \sum_{t\in\mathcal{T}_n} P_\sigma(t)\cdot N(t,b) \\ &\leq \frac{2(n-1)}{b} - 1 \\ &\leq \mathcal{O}\left(\frac{n}{\log n}\right). \end{aligned}$$

Altogether, we get

$$\sum_{t\in\mathcal{T}_n} P_\sigma(t)\cdot|\mathcal{D}_t| \leq \mathcal{O}\left(\frac{n}{\log n}\right).$$

This concludes the proof. □

Corollary 2. *Let $\sigma \in \Sigma^m_{\text{leaf}}$, and let $\mathcal{S} = ((\mathcal{T}_i)_{i\geq 1}, P_\sigma)$ be the corresponding leaf-centric tree source. Then, we have*

$$R_\varnothing(E_{\text{dag}}, \mathcal{S}, i) \leq \mathcal{O}\left(\frac{\log\log i}{\log i}\right).$$

Proof. By Theorem 1 we have

$$\begin{aligned} R_\varnothing(E_{\text{dag}}, \mathcal{S}, i) &\leq C\cdot g\left(\sum_{t\in\mathcal{T}_i} P_\sigma(t)\cdot\frac{|\mathcal{D}_t|}{i}\right) \\ &\leq C\cdot g\left(\frac{1}{\log i}\right) \\ &= C\cdot\frac{\log\log i}{\log i}, \end{aligned}$$

where the constant $C > 0$ only depends on $\mathcal{S}$. This shows the corollary. □

The bound from the above corollary applies in particular to the binary search tree model and the binomial random tree model from Example 5. For the uniform model $\mathcal{S} = ((\mathcal{T}_i)_{i\geq 1}, P_{\sigma_{\text{uni}}})$ we only get the weaker bound using the results on the average size of DAGs in [10]:

$$\begin{aligned} R_\varnothing(E_{\text{dag}}, \mathcal{S}, i) &\leq C\cdot g\left(\sum_{t\in\mathcal{T}_i} P_{\sigma_{\text{uni}}}(t)\cdot\frac{|\mathcal{D}_t|}{i}\right) \\ &\leq C\cdot g\left(\frac{1}{\sqrt{\log i}}\right) \\ &= C\cdot\frac{\log\log i}{\sqrt{\log i}}. \end{aligned}$$

D. Depth-centric binary tree sources

We recall the definition of a natural class of depth-centric tree sources from [28]: Let Σ_{depth} be the set of all mappings $\sigma : \mathbb{N}\times\mathbb{N} \to [0,1]$ such that for all $n \geq 1$:

$$\sum_{i,j\geq 0,\ \max(i,j)=n-1} \sigma(i,j) = 1. \tag{10}$$

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

$$P_\sigma(a) = 1, \tag{11}$$
$$P_\sigma(f(s,t)) = \sigma(d(s),d(t))\cdot P_\sigma(s)\cdot P_\sigma(t). \tag{12}$$

We have $\sum_{t\in\mathcal{T}^n} P_\sigma(t) = 1$ and thus $((\mathcal{T}^i)_{i\geq 0}, P_\sigma)$ is a depth-centric tree source.

The following result is again implicitly shown in [28] and can also be found as a part of the proof of Theorem 6.

Lemma 6. *For every $\sigma \in \Sigma_{\text{depth}}$, the depth-centric tree source $((\mathcal{T}^i)_{i\geq 0}, P_\sigma)$ has the weak domination property.*

We say the mapping $\sigma \in \Sigma_{\text{depth}}$ is *depth-balanced* if there exists a constant c such that for all $(i,j) \in \mathbb{N}\times\mathbb{N}$ with $\sigma(i,j) > 0$ we have $|i-j| \leq c$. In [28], the authors define a condition on σ that is slightly stronger than depth-balancedness, and show that for every such σ, the tree source $((\mathcal{T}^i)_{i\geq 0}, P_\sigma)$ has the representation ratio negligibility property. Similarly to Lemma 3, we will show an even stronger property. To do so, we introduce *β-depth-balanced* trees for $\beta \in \mathbb{N}$. A tree t is called *β-depth-balanced* if for each subtree $f(t_1,t_2)$ of t we have $|d(t_1) - d(t_2)| \leq \beta$. Note that for a depth-balanced mapping $\sigma \in \Sigma_{\text{depth}}$, there is a constant β such that every tree t with $P_\sigma(t) > 0$ is β-depth-balanced. We will use the following lemma:

Lemma 7. *Let $\beta \in \mathbb{N}$ and $c = 1+1/(1+\beta)$ (thus, $1 < c \leq 2$). For every β-depth-balanced binary tree t, we have $|t| \geq c^{d(t)}$.*

Proof. We prove the lemma by induction on $d(t)$. For the only tree $t = a$ of depth $d(t) = 0$, we have $|t| = 1 = c^0$. Consider now a β-depth-balanced tree $t = f(t_1,t_2)$ of depth $d(t) > 0$. We assume $d(t_1) \geq d(t_2)$, the other case is symmetric. Since t is β-depth-balanced, it follows that $d(t_2) \geq d(t_1) - \beta$. To estimate the size $|t| = |t_1| + |t_2|$, we apply the induction hypothesis to t_1 and t_2, which yields

$$\begin{aligned} |t| &= |t_1| + |t_2| \\ &\geq c^{d(t_1)} + c^{d(t_2)} \\ &\geq c^{d(t_1)} + c^{d(t_1)-\beta} \\ &= c^{d(t_1)}\cdot\left(1 + c^{-\beta}\right). \end{aligned}$$

Since $d(t_1)+1 = d(t)$, it only remains to show that $1+c^{-\beta} \geq c$, which can be easily done by induction on $\beta \in \mathbb{N}$. □

Lemma 7 together with results from [11], [13] implies:

Lemma 8. *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 $|\mathcal{D}_t| \leq \alpha \cdot n \cdot \log_2(\log_2 n)/\log_2 n$.*

Proof. If $P_\sigma(t) > 0$, then there exists a constant β such that t and each of its subtrees is β-depth-balanced. By Lemma 7 this implies that every subtree t' has depth at most $c \cdot \log_2 |t'|$ for a constant c that only depends on σ. By [11, Theorem 12][1] (which was implicitly shown in [13]) it follows that there exists a constant α (again, only dependent on σ) such that $|\mathcal{D}_t| \leq \alpha \cdot n \cdot \log_2(\log_2 n)/\log_2 n$. □

Corollary 3. *Let $\sigma \in \Sigma_{depth}$ be a depth-balanced mapping and let $\mathcal{S} = ((\mathcal{T}^i)_{i\geq 0}, P_\sigma)$ be the corresponding depth-centric tree source. Then, we have*

$$R(E_{dag}, \mathcal{S}, i) \leq \mathcal{O}\left(\frac{(\log\log i)^2}{\log i}\right).$$

Proof. Let α be the constant from Lemma 8. Let $t \in \mathcal{T}^i$ such that $P_\sigma(t) > 0$. Lemma 1 and 6 imply

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

Consider the mapping $g(x) = x \cdot \log_2(1/x)$. It is monotonically increasing for $0 \leq x \leq 1/e$. Note that for all $t \in \mathcal{T}^i$ we have $|t| \geq i+1$. Hence, if i is large enough, then Lemma 8 yields for all $t \in \mathcal{T}^i$ with $P_\sigma(t) > 0$ that

$$\begin{aligned} |\mathcal{D}_t|/|t| &\leq \alpha \cdot \log_2(\log_2 |t|)/\log_2 |t| \\ &\leq \log_2(\log_2(i+1))/\log_2(i+1) \leq 1/e. \end{aligned}$$

We obtain

$$\frac{1}{|t|} \cdot (|E_{\text{dag}}(t)| + \log_2(P_\sigma(t))) \leq \mathcal{O}((\log\log i)^2/\log i).$$

This proves the corollary. □

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 4 shows that for every $n \geq 1$, there is a tree $t \in \mathcal{T}_n$ with $|\mathcal{D}_t| = n$. In that case, the bound stated in Lemma 1 cannot be used to show that the worst-case redundancy converges to zero.

[1] We remark that [11, Theorem 12] contains a typing error: "every subtree s of t has depth at most $\alpha \log n + \alpha$" should be replaced by "every subtree s of t has depth at most $\alpha \log |s| + \alpha$".

A. TSLPs in normal form

In this section, we will use TSLPs in a certain normal form, which we introduce first.

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

- $V = \{A_0, A_1, \ldots, A_{n-1}\}$ for some $n \in \mathbb{N}$, $n \geq 1$.
- 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 the word $\rho(A_i) \in (V \cup \{a\})^*$ as follows:

$$\rho(A_i) = \begin{cases} A_j\alpha & \text{if } r(A_i) = A_j(\alpha) \\ A_jA_k & \text{if } r(A_i) = A_j(A_k(x)) \\ \alpha & \text{if } r(A_i) = f(\alpha, x) \text{ or } f(x, \alpha) \end{cases}$$

  Moreover, define the word

$$\rho_{\mathcal{G}} = \rho(A_0)\rho(A_1)\cdots\rho(A_{n-1}),$$

  which belongs to $\{a, A_1, A_2, \ldots, A_{n-1}\}^*$. Then we require that $\rho_{\mathcal{G}}$ has the form $A_1u_1A_2u_2\cdots A_{n-1}u_{n-1}$ with $u_i \in \{a, A_1, A_2, \ldots, A_i\}^*$.
- $\mathsf{val}_{\mathcal{G}}(A_i) \neq \mathsf{val}_{\mathcal{G}}(A_j)$ for $i \neq j$

We define the size of the normal form TSLP $\mathcal{G}$ 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 $\mathcal{G}$. As for DAGs we also allow the TSLP $\mathcal{G}_a = (\{A_0\}, A_0, A_0 \mapsto a)$ in order to get the singleton tree a. In this case, we set $\rho_{\mathcal{G}_a} = \rho(A_0) = a$. It is not hard to show that there is a linear time algorithm that transforms a given TSLP $\mathcal{G}$ into a normal form TSLP $\mathcal{G}'$ that derives the same tree (we will not use this fact). For example, the TSLP from Example 2 is transformed into the normal form TSLP described in Example 7. Let $\mathcal{G} = (V, A_0, r)$ be a TSLP in normal form with $V = \{A_0, A_1, \ldots, A_{n-1}\}$ for the further definitions.

Let $\omega_{\mathcal{G}}$ be the word obtained from $\rho_{\mathcal{G}}$ by removing for every $1 \leq i \leq n-1$ 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 $u_i \in \{a, A_1, A_2, \ldots, A_i\}^*$, 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 7. *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*

$$\begin{aligned} r(A_0) &= A_1(A_2), \\ r(A_1) &= f(x, A_3), \\ r(A_2) &= A_4(A_3), \\ r(A_3) &= A_4(a), \\ r(A_4) &= f(x, a). \end{aligned}$$

Then $\mathcal{G}$ evaluates to $\mathsf{val}(\mathcal{G}) = f(f(f(a,a),a), f(a,a))$. Moreover, we have $\rho_{\mathcal{G}} = A_1A_2A_3A_4A_3A_4aa$ ($u_1 = u_2 = u_3 = \varepsilon$, $u_4 = A_3A_4aa$), $|\mathcal{G}| = 8$ and $\omega_{\mathcal{G}} = A_3A_4aa$.

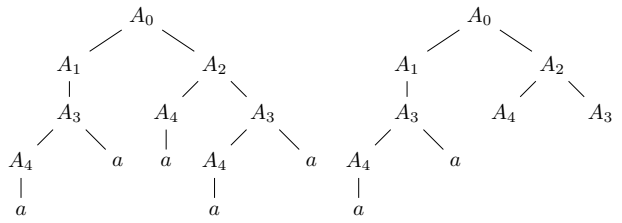

Fig. 3. The derivation tree $T_{\mathcal{G}}$ of the TSLP from Example 7 (left) and an initial subtree T' of $T_{\mathcal{G}}$ (right).

The *derivation tree* $T_{\mathcal{G}}$ of the normal form TSLP $\mathcal{G}$ is a rooted tree, where every node is labelled with a symbol from $V \cup \{a\}$. The root is labelled with A_0. Nodes labelled with a are the leaves of $T_{\mathcal{G}}$. A node v that is labelled with a nonterminal A_i has $|\rho(A_i)|$ many children. If $\rho(A_i) = \alpha \in V_0 \cup \{a\}$ then the single child of v is labelled with α. If $\rho(A_i) = A_j\alpha$ with $\alpha \in V \cup \{a\}$ then the left (resp., right) child of v is labelled with A_j (resp., α). For every node v of $\mathcal{T}_G$ we define the tree or context $t_v = \mathsf{val}_{\mathcal{G}}(\alpha)$ where $\alpha \in V \cup \{a\}$ is the label of v. If $\alpha \in V_0 \cup \{a\}$ then $t_v \in \mathcal{T}$ and if $\alpha \in V_1$ then $t_v \in \mathcal{C}$. The t_v can be also inductively defined by the following rules:

- If u is labelled with a then $t_u = a$.
- If u is labelled with $A_i \in V_1$ and has a single child node v that is labelled with $\alpha \in V_0 \cup \{a\}$, then $t_u = f(t_v, x)$ if $r(A_i) = f(\alpha, x)$ and $t_u = f(x, t_v)$ if $r(A_i) = f(x, \alpha)$ (note that $r(A_i)$ must be of the form $f(x, \alpha)$ or $f(\alpha, x)$ and that $t_v = \mathsf{val}_{\mathcal{G}}(\alpha)$).
- If v has the left child u_1 and the right child u_2, then $t_v = t_{u_1}[t_{u_2}]$ (note that if a node has two children in the derivation tree, then the left child u_1 is labelled with a nonterminal from V_1 and thus t_{u_1} is a context from $\mathcal{C}_V$).

An *initial subtree* of the derivation tree $T_{\mathcal{G}}$ is a tree that can be obtained from $T_{\mathcal{G}}$ as follows: Take a subset U of the nodes of $T_{\mathcal{G}}$ and remove from $T_{\mathcal{G}}$ all proper descendants of nodes from U, i.e., all nodes that are located strictly below a node from U.

Example 8. *Let $\mathcal{G}$ be the normal form TSLP from Example 7. The derivation tree $T_{\mathcal{G}}$ is shown in Figure 3 on the left; an initial subtree T' of it is shown on the right.*

Lemma 9. *Let T' be an initial subtree of $T_{\mathcal{G}}$ and let $v_1, \ldots, v_\ell$ be the sequence of all leaves of T' (in left-to-right order). Then $|\mathsf{val}(\mathcal{G})| = \sum_{i=1}^{\ell} |t_{v_i}|$.*

Proof. Let u be a node of $T_{\mathcal{G}}$ and let T_u be the subtree of $T_{\mathcal{G}}$ rooted in u. Let us first show by induction that $|t_u| = |T_u|$ holds, where as for binary trees, $|T_u|$ denotes the number of leaves of T_u. Recall that $|t_u|$ is the number of a-labelled leaves of t_u; in particular the parameter x is not counted. If u is a leaf of $T_{\mathcal{G}}$ then we have $t_u = a$ and hence $|t_u| = 1 = |T_u|$. If u has two children u_1 and u_2 then $t_u = t_{u_1}[t_{u_2}]$ and we get $|t_u| = |t_{u_1}[t_{u_2}]| = |t_{u_1}| + |t_{u_2}|$ (see Section II-B for the last equality). Hence, we get by induction $|t_u| = |T_{u_1}| + |T_{u_2}| = |T_u|$. Finally, if the node u has a single child v in $T_{\mathcal{G}}$ then either $t_u = f(x, t_v)$ or $t_u = f(t_v, x)$. In both cases we get by induction $|t_u| = |t_v| = |T_v| = |T_u|$.

Let us now conclude the proof of the lemma. Since T' is an initial subtree of $T_{\mathcal{G}}$ we get $|\mathsf{val}(\mathcal{G})| = |T_{\mathcal{G}}| = \sum_{i=1}^{\ell} |T_{v_i}| = \sum_{i=1}^{\ell} |t_{v_i}|$. □

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

$$\gamma_\psi(n) = \max_{t \in \mathcal{T}_n} |\mathcal{G}_t|/n.$$

A key result that we need in this paper is the following theorem from [11]:

Theorem 3. *There exists a grammar-based compressor ψ (working in linear time) with $\gamma_\psi(n) \leq \mathcal{O}(1/\log n)$.*

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 [16] and DAGs [28]. 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 $\mathsf{type}(A_i) \in \{0, 1, 2, 3\}$ of a nonterminal $A_i \in V$ as follows:

$$\mathsf{type}(A_i) = \begin{cases} 0 & \text{if } r(A_i) = A_j(\alpha) \text{ for some } A_j \in V_1 \\ & \quad \text{and } \alpha \in V_0 \cup \{a\} \\ 1 & \text{if } r(A_i) = A_j(A_k(x)) \\ & \quad \text{for some } A_j, A_k \in V_1 \\ 2 & \text{if } r(A_i) = f(\alpha, x) \text{ for some } \alpha \in V_0 \cup \{a\} \\ 3 & \text{if } r(A_i) = f(x, \alpha) \text{ for some } \alpha \in V_0 \cup \{a\} \end{cases}$$

We define the binary word $B(\mathcal{G}) = w_0 w_1 w_2 w_3 w_4$, where the words $w_i \in \{0,1\}^*$, $0 \leq i \leq 4$, are defined as follows:

- $w_0 = 1^{n-1}0$,
- $w_1 = a_0 b_0 a_1 b_1 \cdots a_{n-1} b_{n-1}$, where $a_j b_j$ is the 2-bit binary encoding of $\mathsf{type}(A_j)$,
- Assume that $\rho_{\mathcal{G}} = A_1 u_1 A_2 u_2 \cdots A_{n-1} u_{n-1}$ with $u_i \in \{a, A_1, \ldots, A_i\}^*$. Then $w_2 = 1^{|u_1|}01^{|u_2|}0 \cdots 1^{|u_{n-1}|}0$, which is ε in case $n = 1$. Note that $|w_2| = m$.
- For $1 \leq i \leq n-1$ let $k_i = |\rho_{\mathcal{G}}|_{A_i} \geq 1$ be the number of occurrences of the nonterminal A_i in the word $\rho_{\mathcal{G}}$. Then $w_3 = 1^{k_1 - 1}01^{k_2 - 1}0 \cdots 1^{k_{n-1} - 1}0$, which is ε in case $n = 1$.
- The word w_4 encodes the word $\omega_{\mathcal{G}}$ using the well-known enumerative encoding [5]. 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,
$$|S| = \frac{(m-n+1)!}{\eta(a)! \cdot \prod_{i=1}^{n-1} \eta(A_i)!}. \tag{13}$$
Let $v_0, v_1, \ldots, v_{|S|-1}$ be the lexicographic enumeration of the words from S with respect to the alphabet order $a, A_1, \ldots, A_{n-1}$. Then w_4 is the binary encoding of the unique index 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 the length $\lceil \log_2 |S| \rceil$).

Example 9. *Consider the normal form TSLP $\mathcal{G}$ from Example 7. We have $w_0 = 11110$, $w_1 = 0011000011$, $w_2 = 00011110$ and $w_3 = 001010$. 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 thus $w_4 = 1000$.*

Lemma 10. *The set of code words $B(\mathcal{G})$, where $\mathcal{G}$ ranges over all TSLPs in normal form, is a prefix code.*

Proof. Let $B(\mathcal{G}) = w_0w_1w_2w_3w_4$ with w_i defined as above. We show how to recover the TSLP $\mathcal{G}$. From $B(\mathcal{G})$ and the fact that $w_0 = 1^{n-1}0$ we can compute $n = |V|$ and the suffix $w_1w_2w_3w_4$. Since $|w_1| = 2n$ we can then determine w_1 and the suffix $w_2w_3w_4$. The word w_1 encodes the type of every nonterminal. Since $w_2 = 1^{|u_1|}01^{|u_2|}0 \cdots 1^{|u_{n-1}|}0$ and we know n we can compute from $w_2w_3w_4$ the word w_2 and the suffix w_3w_4. The word w_2 allows to compute the positions where we have deleted the nonterminals $A_1, A_2, \ldots, A_{n-1}$ (in that order) from $\rho_{\mathcal{G}}$ during the computation of $\omega_{\mathcal{G}}$. Hence, in order to compute $\rho_{\mathcal{G}}$, one only needs $\omega_{\mathcal{G}}$. Since $w_3 = 1^{k_1-1}01^{k_2-1}0 \cdots 1^{k_{n-1}-1}0$ and we know n, we can compute w_3 and w_4. The word w_3 determines the frequencies $\eta(a), \eta(A_1), \ldots, \eta(A_{n-1})$ of the symbols in $\omega_{\mathcal{G}}$. Using these frequencies one computes the size $|S|$ from (13) and hence the length $\lceil \log_2 |S| \rceil$ of w_4. From w_4, one can then compute $\omega_{\mathcal{G}}$. Finally, $\rho_{\mathcal{G}}$ together with the types of the nonterminals (which are encoded by w_1) completely determines $\mathcal{G}$. The argument shows also that $B(\mathcal{G})$ cannot be a prefix of $B(\mathcal{G}')$ for different normal form TSLPs $\mathcal{G}$ and $\mathcal{G}'$. □

Note that $|B(\mathcal{G})| \leq \mathcal{O}(|\mathcal{G}|) + |w_4|$. By using the well-known bound on the code length of enumerative encoding [6, Theorem 11.1.3], we get:

Lemma 11. *For the length of the binary coding $B(\mathcal{G})$ we have: $|B(\mathcal{G})| \leq \mathcal{O}(|\mathcal{G}|) + H(\mathcal{G})$.*

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 Section II-B. We say that $((\mathcal{F}_i)_{i\in\mathbb{N}}, P)$ has the *strong domination property* if there exists a mapping $\lambda : \mathcal{T} \cup \mathcal{C} \to \mathbb{R}_{>0}$ with the following properties:

(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 $n \geq 1$.

Recall the definition of the trees/contexts t_u for a node u of a derivation tree of a normal form TSLP from Section IV-A.

Lemma 12. *Let $\lambda : \mathcal{T} \cup \mathcal{C} \to \mathbb{R}_{>0}$ be a mapping that satisfies the properties* (ii) *and* (iii) *of the strong domination property. Let $\mathcal{G} = (V, A_0, r)$ be a TSLP in normal form with $\mathsf{val}(\mathcal{G}) = t$ and let T' be an initial subtree of the derivation tree $T_{\mathcal{G}}$. Let $v_1, \ldots, v_\ell$ be the sequence of leaves of T'. Then $\lambda(t) \leq \prod_{i=1}^{\ell} \lambda(t_{v_i})$.*

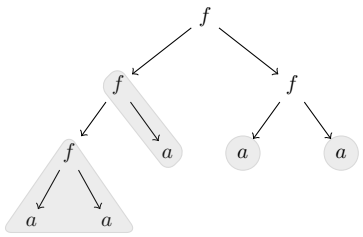

Fig. 4. The tree $\mathsf{val}(\mathcal{G})$ of the TSLP from Example 7. The canonical occurrences of the trees/contexts in $(a, a, \mathsf{val}_{\mathcal{G}}(A_4), \mathsf{val}_{\mathcal{G}}(A_3)) = (a, a, f(x,a), f(a,a))$ used in the proof of Lemma 12 are highlighted.

Proof. Consider a node u of the derivation tree $T_{\mathcal{G}}$. The tree/context t_u clearly occurs in t. One can define a canonical occurrence of t_u in t which is identified with a set V_u of nodes of t. For the root r, V_r is the set of all nodes of t. Now consider a node u of $T_{\mathcal{G}}$ for which V_u has been defined. If u has two children u_1 and u_2 then $t_u = t_{u_1}[t_{u_2}]$. Then V_{u_1} and V_{u_2} can be uniquely defined by the following conditions: there is a node $y \in V_u$ such that V_{u_2} contains all nodes $z \in V_u$ that are descendants of y in the tree t (including y), $V_{u_1} = V_u \setminus V_{u_2}$ and the nodes in V_{u_i} induce an occurrence of t_i in t. If the node u has a single child v in $T_{\mathcal{G}}$ then either $t_u = f(x, t_v)$ or $t_u = f(t_v, x)$. Let $y \in V_u$ be the root node of V_u. If $t_u = f(t_v, x)$ then V_v is the subtree of t rooted in the left child of y and if $t_u = f(x, t_v)$ then V_v is the subtree of t rooted in the right child of y.

Now consider the nodes $v_1, \ldots, v_\ell$ from the lemma. Figure 4 shows the node sets $V_{v_1}, V_{v_2}, V_{v_3}, V_{v_4}$ for the four leaf nodes of the initial subtree T' from Figure 3. Since these nodes are pairwise incomparable with respect to the ancestor relation of $T_{\mathcal{G}}$, the node sets V_{v_i} are pairwise disjoint. This allows us to prove $\lambda(t) \leq \prod_{i=1}^{\ell} \lambda(t_{v_i})$ inductively. The case that $t = a$ is clear. Now assume that $t = f(t_1, t_2)$. First assume that the root node of t does not belong to some of the sets V_{v_i}. Then every set V_{v_i} is either contained in the left subtree t_1 or in the right subtree t_2. W.l.o.g. assume that $V_{v_1}, \ldots, V_{v_k}$ are contained in t_1 and $V_{v_{k+1}}, \ldots, V_{v_\ell}$ are contained in t_2. By induction and condition (ii) of the strong domination property we get

$$\lambda(t) \leq \lambda(t_1) \cdot \lambda(t_2) \leq \prod_{i=1}^{k} \lambda(t_{v_i}) \cdot \prod_{i=k+1}^{\ell} \lambda(t_{v_i}).$$

Now assume that the root node of t belongs to a set V_{v_i}. W.l.o.g. assume that $i = 1$. If t_{v_1} is a tree then we must have $t = t_{v_1}$ and $\ell = 1$ and the statement of the lemma holds. Otherwise, t_{v_1} is a context, $t = t_{v_1}[t']$ and all $V_{v_2}, \ldots, V_{v_\ell}$ are contained in t'. By induction and condition (iii) of the strong domination property we get

$$\lambda(t) \leq \lambda(t_{v_1}) \cdot \lambda(t') \leq \lambda(t_{v_1}) \cdot \prod_{i=2}^{\ell} \lambda(t_{v_i}).$$

This concludes the proof of the lemma. □

The proof of the following lemma combines ideas from [16] and [28].

Lemma 13. *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*

$$H(\mathcal{G}) \leq -\log_2 P(t) + \mathcal{O}(|\mathcal{G}|) + \mathcal{O}(|\mathcal{G}| \cdot \log_2(n/|\mathcal{G}|)).$$

Proof. Let $m = |\mathcal{G}| = |\rho_{\mathcal{G}}|$ be the size of $\mathcal{G}$, $k = |V|$, and $\ell := m + 1 - k \leq m$. Let $T = T_{\mathcal{G}}$ be the derivation tree of $\mathcal{G}$. We define an initial subtree T' as follows: we walk in preorder (depth-first, left-to-right) over the tree T. Every time we visit a non-leaf node v in T that is labelled with a nonterminal that has been seen before during the preorder traversal, we remove from T all proper descendants of v. Thus, for every $A_i \in V$ there is exactly one non-leaf node in T' that is labelled with A_i. For the TSLP from Example 7, the tree T' is shown in Figure 3 on the right.

Recall the definition of the words $\rho_{\mathcal{G}}$ and $\omega_{\mathcal{G}}$ from Section IV-A. The word $\rho_{\mathcal{G}}$ can be obtained by writing down for every node v of T the labels of v's children. Moreover, the word $\omega_{\mathcal{G}}$ is obtained by writing down (in the right order) the labels of the leaves of T'. Note that T' has exactly $m + 1$ many nodes and k non-leaf nodes. Thus, T' has ℓ leaves. Let $v_1, v_2, \ldots, v_\ell$ be the sequence of all leaves of T' (w.l.o.g. in preorder) and let $\alpha_i \in \{a, A_1, \ldots, A_{k-1}\}$ be the label of v_i. Let $\overline{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_\ell)$. Then $\overline{\alpha}$ is a permutation of $\omega_{\mathcal{G}}$. We therefore have $|\omega_{\mathcal{G}}|_\alpha = |\overline{\alpha}|_\alpha$ for every $\alpha \in \{a, A_1, \ldots, A_{k-1}\}$. Hence, the empirical distributions $p_{\overline{\alpha}}$ and $p_{\omega_{\mathcal{G}}}$ are identical. For the TSLP from Example 7 we get $\overline{\alpha} = (a, a, A_4, A_3)$ (this is the sequence of labels of the leaf nodes for the tree in Figure 3 on the right). Let $t_i = \mathsf{val}_{\mathcal{G}}(\alpha_i) \in \mathcal{T} \cup \mathcal{C}$. Since $\mathsf{val}_{\mathcal{G}}(A_i) \neq \mathsf{val}_{\mathcal{G}}(A_j)$ for all $i \neq j$ and $\mathsf{val}_{\mathcal{G}}(A_i) \neq a$ for all i (this holds for every normal form TSLP that produces a tree of size at least two), the tuple $\overline{t} = (t_1, t_2, \ldots, t_\ell)$ satisfies $p_{\omega_{\mathcal{G}}}(\alpha_i) = p_{\overline{t}}(t_i)$ for all $1 \leq i \leq \ell$. For the TSLP from Example 7 we get $\overline{t} = (a, a, \mathsf{val}_{\mathcal{G}}(A_4), \mathsf{val}_{\mathcal{G}}(A_3)) = (a, a, f(x, a), f(a, a))$ (see Figure 4).

Lemma 9 yields

$$\sum_{i=1}^{l} |t_i| = |t| = n \tag{14}$$

since $t \in \mathcal{T}_n$, whereas Lemma 12 yields

$$\lambda(t) \leq \prod_{i=1}^{\ell} \lambda(t_i). \tag{15}$$

For $j \in \mathbb{N}$, $j \geq 1$, let

$$M_j = \sum_{u \in \mathcal{T}_j \cup \mathcal{C}_j} \lambda(u) \leq c_1 \cdot j^{c_2},$$

where c_1 and c_2 are the constants from condition (iv) of the strong domination property. Let $D := 6/\pi^2 \geq 1/2$ and define for every $u \in \mathcal{T}_j \cup \mathcal{C}_j$:

$$q(u) := \frac{D \cdot \lambda(u)}{M_j \cdot j^2} > 0. \tag{16}$$

We get

$$\sum_{j\geq 1} \sum_{u \in \mathcal{T}_j \cup \mathcal{C}_j} q(u) = D \cdot \sum_{j \geq 1} \frac{1}{j^2} = 1.$$

Hence, we have $q(t_1) + q(t_2) + \cdots + q(t_\ell) \leq 1$. Using Shannon's inequality (3) we get

$$\begin{aligned} H(\mathcal{G}) &= H(\omega_{\mathcal{G}}) \\ &= \sum_{i=1}^{\ell} -\log_2 p_{\omega_{\mathcal{G}}}(\alpha_i) \\ &= \sum_{i=1}^{\ell} -\log_2 p_{\overline{t}}(t_i) \leq \sum_{i=1}^{\ell} -\log_2 q(t_i). \end{aligned}$$

Using (16) and $D \geq 1/2$ we obtain

$$\begin{aligned} H(\mathcal{G}) &\leq \sum_{i=1}^{\ell} -\log_2 \left(\frac{D \cdot \lambda(t_i)}{M_{|t_i|} \cdot |t_i|^2} \right) \\ &= -\ell \cdot \log_2 D - \sum_{i=1}^{\ell} \log_2 \lambda(t_i) \\ &\quad + \sum_{i=1}^{\ell} \log_2 M_{|t_i|} + 2 \sum_{i=1}^{\ell} \log_2 |t_i| \\ &\leq \ell - \sum_{i=1}^{\ell} \log_2 \lambda(t_i) + \sum_{i=1}^{\ell} (\log_2 c_1 + c_2 \log_2 |t_i|) \\ &\quad + 2 \sum_{i=1}^{\ell} \log_2 |t_i| \\ &= (1 + \log_2 c_1) \cdot \ell - \sum_{i=1}^{\ell} \log_2 \lambda(t_i) \\ &\quad + (2 + c_2) \cdot \sum_{i=1}^{\ell} \log_2 |t_i|. \end{aligned}$$

From (15) and condition (i) of the strong domination property we get

$$\sum_{i=1}^{\ell} \log_2 \lambda(t_i) \geq \log_2 \lambda(t) \geq \log_2 P(t).$$

Moreover, Jensen's inequality and (14) gives

$$\sum_{i=1}^{\ell} \log_2 |t_i| \leq \ell \cdot \log_2 \left(\frac{1}{\ell} \cdot \sum_{i=1}^{\ell} |t_i| \right) = \ell \cdot \log_2(n/\ell).$$

With $\ell \leq m$ we obtain

$$\begin{aligned} H(\mathcal{G}) &\leq (1 + \log_2 c_1) \cdot \ell - \log_2 P(t) \\ &\quad + (2 + c_2) \cdot \ell \cdot \log_2(n/\ell) \\ &\leq -\log_2 P(t) + (1 + \log_2 c_1) \cdot m \\ &\quad + (2 + c_2) \cdot m \cdot \log_2(n/m). \end{aligned}$$

This shows the lemma. □

Let $\psi : t \mapsto \mathcal{G}_t$ be a grammar-based tree compressor. We then consider the tree encoder $E_\psi : \mathcal{T} \to \{0,1\}^*$ defined by $E_\psi(t) = B(\mathcal{G}_t)$. Recall the definition of the worst-case redundancy $R(E_\psi, \mathcal{S}, i)$ from Section II-B.

Theorem 4. *Assume that $\mathcal{S} = ((\mathcal{F}_i)_{i\in\mathbb{N}}, P)$ has the strong domination property. Let ψ be a grammar-based compressor such that $\gamma_\psi(n) \le \gamma(n)$ for a monotonically decreasing function $\gamma(n)$ with $\lim_{n\to\infty}\gamma(n) = 0$. Let $n_i = \min\{|t| \mid t \in \mathcal{F}_i\}$ and assume that $n_i < n_{i+1}$ for all $i \in \mathbb{N}$.[2] Then, we have*

$$R(E_\psi, \mathcal{S}, i) \le \mathcal{O}(\gamma(n_i) \cdot \log_2(1/\gamma(n_i))).$$

Proof. Let $t \in \mathcal{T}$ and $\psi(t) = \mathcal{G}_t$. With Lemma 11 and 13 we get

$$\begin{aligned} & \frac{1}{|t|} \cdot (|B(\mathcal{G}_t)| + \log_2 P(t)) \\ \le\ & \frac{1}{|t|} \cdot (H(\mathcal{G}_t) + \mathcal{O}(|\mathcal{G}_t|) + \log_2 P(t)) \\ \le\ & \frac{1}{|t|} \cdot (\mathcal{O}(|\mathcal{G}_t|) + \mathcal{O}(|\mathcal{G}_t| \cdot \log_2(|t|/|\mathcal{G}_t|)) \\ =\ & \mathcal{O}(|\mathcal{G}_t|/|t|) + \mathcal{O}(|\mathcal{G}_t|/|t| \cdot \log_2(|t|/|\mathcal{G}_t|)). \end{aligned}$$

Consider the mapping g with $g(x) = x \cdot \log_2(1/x)$. It is monotonically increasing for $0 \le x \le 1/e$. If i is large enough, we have for all $t \in \mathcal{F}_i$ that

$$|\mathcal{G}_t|/|t| \le \gamma_\psi(|t|) \le \gamma(|t|) \le \gamma(n_i) \le 1/e.$$

Hence, we get

$$\begin{aligned} |\mathcal{G}_t|/|t| \cdot \log_2(|t|/|\mathcal{G}_t|) &= g(|\mathcal{G}_t|/|t|) \\ &\le g(\gamma(n_i)) \\ &= \gamma(n_i) \cdot \log_2(1/\gamma(n_i)). \end{aligned}$$

This implies

$$\begin{aligned} R(E_\psi, \mathcal{S}, i) &= \max_{t\in\mathcal{F}_i, P(t)>0} \frac{1}{|t|} \cdot (|B(\mathcal{G}_t)| + \log_2 P(t)) \\ &\le \mathcal{O}(\gamma(n_i)) + \mathcal{O}(\gamma(n_i) \cdot \log(1/\gamma(n_i))) \\ &= \mathcal{O}(\gamma(n_i) \cdot \log(1/\gamma(n_i))), \end{aligned}$$

which proves the theorem. □

Note that the minimal size of a tree in $\mathcal{T}_{i+1}$ (resp. $\mathcal{T}^i$) is $i+1$. Hence, Theorem 3 and 4 yield:

Corollary 4. *There exists a grammar-based tree compressor ψ (working in linear time) such that $R(E_\psi, \mathcal{S}, i) \le \mathcal{O}(\log\log i/\log i)$ for every leaf-centric or depth-centric tree source $\mathcal{S}$ having the strong domination property.*

In the rest of the paper, we will present classes of leaf-centric and depth-centric tree sources that have the strong domination property.

D. Leaf-centric binary tree sources

Recall the definition of the class of mappings Σ_{leaf} by equations (4), (5), and (6) in Section III-C and the corresponding class of leaf-centric tree sources. In this section, we state a condition on the mapping $\sigma \in \Sigma_{\text{leaf}}$ that enforces the strong domination property for the leaf-centric tree source $((\mathcal{T}_i)_{i\ge1}, P_\sigma)$. This allows to apply Corollary 4.

[2]This is the case for leaf-centric and depth-centric tree sources.

Theorem 5. *If $\sigma \in \Sigma_{\text{leaf}}$ satisfies*

$$\sigma(i,j) \ge \sigma(i,j+1) \text{ and } \sigma(i,j) \ge \sigma(i+1,j) \tag{17}$$

for all $i, j \ge 1$, then $((\mathcal{T}_i)_{i\ge1}, P_\sigma)$ has the strong domination property.

Proof. First, we naturally extend P_σ to a context $t \in \mathcal{C}$ using equations (5) and (6), where we set $\sigma(0,k) = \sigma(k,0) = 1$ for all $k \ge 1$ and $P_\sigma(x) = 1$. Note that $|x| = 0$. Also note that P_σ is not a probability distribution on $\mathcal{C}_n$. We have for instance $\sum_{t\in\mathcal{C}_1} P_\sigma(t) = P_\sigma(f(x,a)) + P_\sigma(f(a,x)) = 2$. Let $B_n = |\mathcal{T}_n| = C_{n-1}$ (the $(n-1)^{\text{th}}$ Catalan number) for $n \ge 1$. We set $B_0 = 1$ and define $\lambda : \mathcal{T} \cup \mathcal{C} \to \mathbb{R}_{>0}$ by

$$\lambda(t) = \max\left\{\frac{1}{B_{|t|}}, P_\sigma(t)\right\}.$$

We show the four points from the strong domination property for the mapping λ.

The first point of the strong domination property, i.e., $\lambda(t) \ge P_\sigma(t)$ for all $t \in \mathcal{T}$, is obviously true. We now prove the second point, i.e., $\lambda(f(s,t)) \le \lambda(s) \cdot \lambda(t)$ for all $s, t \in \mathcal{T}$. Assume first that $\lambda(f(s,t)) = 1/B_{|s|+|t|}$. The inequality $B_{m+k} \ge B_m \cdot B_k$ for all $m, k \ge 0$ yields

$$\frac{1}{B_{|s|+|t|}} \le \frac{1}{B_{|s|}} \cdot \frac{1}{B_{|t|}} \le \lambda(s) \cdot \lambda(t). \tag{18}$$

On the other hand, if $\lambda(f(s,t)) = P_\sigma(f(s,t))$, then we have

$$\begin{aligned} P_\sigma(f(s,t)) &= \sigma(|s|,|t|) \cdot P_\sigma(s) \cdot P_\sigma(t) \\ &\le P_\sigma(s) \cdot P_\sigma(t) \\ &\le \lambda(s) \cdot \lambda(t), \end{aligned}$$

since $0 \le \sigma(i,j) \le 1$ for all i, j.

We now consider the third point, i.e., $\lambda(s[t]) \le \lambda(s) \cdot \lambda(t)$ for all $s \in \mathcal{C}$ and $t \in \mathcal{T}$. Note first that the case $\lambda(s[t]) = 1/B_{|s[t]|}$ follows again from equation (18), since $|s[t]| = |s| + |t|$. So we assume that $\lambda(s[t]) = P_\sigma(s[t])$. Let k be the length (measured in the number of edges) from the root of s to the unique x-labelled node in s.

We show $P_\sigma(s[t]) \le P_\sigma(s) \cdot P_\sigma(t)$ by induction over $k \ge 0$. If $k = 0$ then $s = x$ and we get $P_\sigma(s[t]) = P_\sigma(t) = P_\sigma(x) \cdot P_\sigma(t) = P_\sigma(s) \cdot P_\sigma(t)$. Let us now assume that $k \ge 1$. Then, s must have the form $s = f(u,v)$. Without loss of generality assume that x occurs in u; the other case is of course symmetric. Therefore, we have $s[t] = f(u[t], v)$. The tree $u[t]$ fulfills the induction hypothesis and therefore $P_\sigma(u[t]) \le P_\sigma(u) \cdot P_\sigma(t)$. Moreover, we have $\sigma(|u[t]|, |v|) = \sigma(|u| + |t|, |v|) \le \sigma(|u|, |v|)$. We get

$$\begin{aligned} P_\sigma(s[t]) &= P_\sigma(f(u[t], v)) \\ &= P_\sigma(u[t]) \cdot \sigma(|u[t]|, |v|) \cdot P_\sigma(v) \\ &\le P_\sigma(t) \cdot P_\sigma(u) \cdot \sigma(|u|, |v|) \cdot P_\sigma(v) \\ &= P_\sigma(t) \cdot P_\sigma(s). \end{aligned}$$

For the fourth property we show $\sum_{t\in\mathcal{T}_n\cup\mathcal{C}_n} \lambda(t) \le 8n - 2$ for all $n \ge 1$. First, we have

$$\sum_{t\in\mathcal{T}_n} \lambda(t) \le \sum_{t\in\mathcal{T}_n} \left(B_n^{-1} + P_\sigma(t)\right) = 2. \tag{19}$$

We show that $\sum_{t\in\mathcal{C}_n}\lambda(t)\le 8n-4$. Every tree $t\in\mathcal{T}_n$ has $2n-1$ nodes. Let v be a node of t and t_v the subtree of t rooted at v. We obtain two contexts from t and v by replacing in t the subtree t_v by either $f(x,t_v)$ or $f(t_v,x)$. Let us denote the resulting contexts by $t_{v,1}$ and $t_{v,2}$. We have $\lambda(t_{v,1}) = \lambda(t_{v,2}) = \lambda(t)$ for every node v of t. Moreover, for every context $t\in\mathcal{C}_n$ there exists a tree $t'\in\mathcal{T}_n$ and a node v of t' such that $t'_{v,1}=t$ or $t'_{v,2}=t$ (depending on whether x is the left or right child of its parent node in t). Hence, we have

$$\sum_{t\in\mathcal{C}_n}\lambda(t)\le(4n-2)\cdot\sum_{t\in\mathcal{T}_n}\lambda(t). \tag{20}$$

With equation (19) we get $\sum_{t\in\mathcal{T}_n\cup\mathcal{C}_n}\lambda(t)\le 8n-2$. □

Example 10. *Let us come back to the three leaf-centric tree sources from Example 5. The mappings σ_{bst} and σ_{uni} satisfy the condition from* (17)*. Hence, the grammar-based compressor E_ψ achieves a worst-case redundancy of $\mathcal{O}(\log\log i/\log i)$ for the binary search tree model and the uniform model by Corollary 4. Recall that for the DAG-based compressor E_{dag} we only get an average-case redundancy of $\mathcal{O}(\log\log i/\log i)$ for the binary search tree model by Corollary 2. For the uniform model, E_{dag} yields an average-case redundancy of $\mathcal{O}(\log\log i/\sqrt{\log i})$; see the remark after Corollary 2. The mapping σ_{bin} does not satisfy condition* (17)*.*

E. Depth-centric binary tree sources

Recall the definition of the class of mappings Σ_{depth} by equations (10), (11), and (12) in Section III-D, and the corresponding class of depth-centric tree sources. In this section, we state a condition on the mapping $\sigma\in\Sigma_{\text{depth}}$ that enforces the strong domination property for the depth-centric tree source $((\mathcal{T}_i)_{i\ge1},P_\sigma)$. This allows again to apply Corollary 4.

Theorem 6. *If $\sigma\in\Sigma_{depth}$ satisfies $\sigma(i,j)\ge\sigma(i,j+1)$ and $\sigma(i,j)\ge\sigma(i+1,j)$ for all $i,j\ge0$, then $((\mathcal{T}^i)_{i\ge0},P_\sigma)$ has the strong domination property.*

Proof. Recall that the depth of a context $t\in\mathcal{C}$ is defined as the depth of the tree $t[a]$. Using this information, we extend P_σ to a context $t\in\mathcal{C}$ using equations (11) and (12), where we set $P_\sigma(x)=1$. Similarly to the proof of Theorem 5 for leaf-centric tree sources, we define

$$\lambda(t)=\max\left\{\frac{1}{B_{|t|}},P_\sigma(t)\right\}.$$

The first point of the strong domination property, i.e., $\lambda(t)\ge P_\sigma(t)$ for all $t\in\mathcal{T}$, follows directly from the definition of λ. Now we prove the second point, i.e., $\lambda(f(s,t))\le\lambda(s)\cdot\lambda(t)$ for all $s,t\in\mathcal{T}$. The case $\lambda(f(s,t))=1/B_{|s|+|t|}$ is covered by equation (18). If otherwise $\lambda(f(s,t))=P_\sigma(f(s,t))$, then

$$\begin{aligned}P_\sigma(f(s,t)) &= \sigma(d(s),d(t))\cdot P_\sigma(s)\cdot P_\sigma(t)\\ &\le P_\sigma(s)\cdot P_\sigma(t)\\ &\le \lambda(s)\cdot\lambda(t),\end{aligned}$$

since $0\le\sigma(i,j)\le1$ for all i,j.

Consider now the third point, i.e., $\lambda(s[t])\le\lambda(s)\cdot\lambda(t)$ for all $s\in\mathcal{C}$ and $t\in\mathcal{T}$. Note that the case $\lambda(s[t])=1/B_{|s[t]|}$ follows again from equation (18). Hence, we can assume that $\lambda(s[t])=P_\sigma(s[t])$. Again, we prove $P_\sigma(s[t])\le P_\sigma(s)\cdot P_\sigma(t)$ by induction over the length $k\ge0$ (measured in the number of edges) from the root of s to the unique x-labelled node in s. If $k=0$ then $s=x$ and $P_\sigma(s)=1$, which gives us $P_\sigma(s[t])=P_\sigma(t)=P_\sigma(s)\cdot P_\sigma(t)$. We now assume $k\ge1$ and $s=f(u,v)$. Without loss of generality assume that x occurs in u; the other case is symmetric. Therefore, we have $s[t]=f(u[t],v)$. We apply the induction hypothesis to the tree $u[t]$, which yields $P_\sigma(u[t])\le P_\sigma(u)\cdot P_\sigma(t)$. Moreover, since $d(u)\le d(u[t])$ we have $\sigma(d(u[t]),d(v))\le\sigma(d(u),d(v))$. It follows that

$$\begin{aligned}P_\sigma(s[t]) &= P_\sigma(f(u[t],v))\\ &= P_\sigma(u[t])\cdot\sigma(d(u[t]),d(v))\cdot P_\sigma(v)\\ &\le P_\sigma(t)\cdot P_\sigma(u)\cdot\sigma(d(u),d(v))\cdot P_\sigma(v)\\ &= P_\sigma(t)\cdot P_\sigma(s).\end{aligned}$$

For the fourth property we show

$$\sum_{t\in\mathcal{T}_n\cup\mathcal{C}_n}\lambda(t)\le4n^2+3n-1$$

for all $n\ge1$. First, we have

$$\sum_{t\in\mathcal{T}_n}\lambda(t)\le\sum_{t\in\mathcal{T}_n}\left(B_n^{-1}+P_\sigma(t)\right)=1+\sum_{t\in\mathcal{T}_n}P_\sigma(t). \tag{21}$$

Note that P_σ is not a probability distribution on $\mathcal{T}_n$ since this section deals with depth-centric tree sources. But for each tree $t\in\mathcal{T}_n$ we have $\lceil\log_2(n)\rceil\le d(t)\le n-1$, which yields

$$\sum_{t\in\mathcal{T}_n}P_\sigma(t)\le\sum_{i=\lceil\log_2(n)\rceil}^{n-1}\sum_{t\in\mathcal{T}^i}P_\sigma(t)=n-\lceil\log_2(n)\rceil\le n.$$

Together with equation (21) we get $\sum_{t\in\mathcal{T}_n}\lambda(t)\le n+1$. The remaining part $\sum_{t\in\mathcal{C}_n}\lambda(t)$ can be estimated with help of equation (20) from the corresponding part in the proof of Theorem 5. In total, we have

$$\begin{aligned}\sum_{t\in\mathcal{T}_n\cup\mathcal{C}_n}\lambda(t) &\le (4n-1)\sum_{t\in\mathcal{T}_n}\lambda(t)\\ &\le (4n-1)(n+1)\\ &= 4n^2+3n-1.\end{aligned}$$

This proves the theorem. □

V. Future research

We plan to investigate, whether the strong domination property can be shown also for other classes of tree sources. An interesting class are the tree sources derived from stochastic context-free grammars [24]. Another interesting question is, whether the convergence rate of $\mathcal{O}(\log\log i/\log i)$ in Corollary 4 can be improved to $\mathcal{O}(1/\log i)$. In the context of grammar-based string compression, such an improvement has been accomplished in [17].

Acknowledgment. We thank the anonymous reviewers for their comments which revealed a small mistake in the conference version.

REFERENCES

[1] Janos Aczél. On Shannon's inequality, optimal coding, and characterizations of Shannon's and Renyi's entropies. Technical Report Research Report AA-73-05, University of Waterloo, 1973. https://cs.uwaterloo.ca/research/tr/1973/CS-73-05.pdf.

[2] Mireille Bousquet-Mélou, Markus Lohrey, Sebastian Maneth, and Eric Noeth. XML compression via DAGs. *Theory of Computing Systems*, 57(4):1322–1371, 2015.

[3] Peter Brass. *Advanced Data Structures*. Cambridge University Press, New York, NY, USA, 1 edition, 2008.

[4] Yongwook Choi and Wojciech Szpankowski. Compression of graphical structures: Fundamental limits, algorithms, and experiments. *IEEE Transactions on Information Theory*, 58(2):620–638, 2012.

[5] Thomas M. Cover. Enumerative source encoding. *IEEE Transactions on Information Theory*, 19(1):73–77, 1973.

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

[7] Luc Devroye. On the richness of the collection of subtrees in random binary search trees. *Inf. Process. Lett.*, 65(4):195–199, 1998.

[8] Philippe Flajolet, Xavier Gourdon, and Conrado Martinez. Patterns in random binary search trees. *Random Struct. Algorithms*, 11(3):223–244, 1997.

[9] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.

[10] Philippe Flajolet, Paolo Sipala, and Jean-Marc Steyaert. Analytic variations on the common subexpression problem. In *Proceedings of the 17th International Colloquium on Automata, Languages and Programming (ICALP 1990)*, volume 443 of *Lecture Notes in Computer Science*, pages 220–234. Springer, 1990.

[11] Moses Ganardi, Danny Hucke, Artur Jeż, Markus Lohrey, and Eric Noeth. Constructing small tree grammars and small circuits for formulas. *Journal of Computer and System Sciences*, 86:136–158, 2017.

[12] John E. Littlewood Godfrey H. Hardy and George Pólya. *Inequalities*. Cambridge University Press, 1952.

[13] Lorenz Hübschle-Schneider and Rajeev Raman. Tree compression with top trees revisited. In *Proceedings of the 14th International Symposium on Experimental Algorithms, SEA 2015*, volume 9125 of *Lecture Notes in Computer Science*, pages 15–27. Springer, 2015.

[14] Danny Hucke and Markus Lohrey. Universal tree source coding using grammar-based compression. In *Proceedings of the 2017 IEEE International Symposium on Information Theory, ISIT 2017*, pages 1753–1757. IEEE, 2017.

[15] John C. Kieffer. A survey of Bratteli information source theory. In *Proceedings of the IEEE International Symposium on Information Theory, ISIT 2016*, pages 16–20. IEEE, 2016.

[16] John C. Kieffer and En-Hui Yang. Grammar-based codes: A new class of universal lossless source codes. *IEEE Transactions on Information Theory*, 46(3):737–754, 2000.

[17] John C. Kieffer and En-Hui Yang. Structured grammar-based codes for universal lossless data compression. *Communications in Information and Systems*, 2(1):29–52, 2002.

[18] John C. Kieffer, En-Hui Yang, Gregory J. Nelson, and Pamela C. Cosman. Universal lossless compression via multilevel pattern matching. *IEEE Transactions on Information Theory*, 46(4):1227–1245, 2000.

[19] John C. Kieffer, En-Hui Yang, and Wojciech Szpankowski. Structural complexity of random binary trees. In *Proceedings of the IEEE International Symposium on Information Theory, ISIT 2009*, pages 635–639. IEEE, 2009.

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

[21] Markus Lohrey. Grammar-based tree compression. In *Proceedings of the 19th International Conference on Developments in Language Theory, DLT 2015*, volume 9168 of *Lecture Notes in Computer Science*, pages 46–57. Springer, 2015.

[22] Markus Lohrey, Sebastian Maneth, and Roy Mennicke. XML tree structure compression using RePair. *Inf. Syst.*, 38(8):1150–1167, 2013.

[23] Abram Magner, Krzysztof Turowski, and Wojciech Szpankowski. Lossless compression of binary trees with correlated vertex names. In *Proceedings of the IEEE International Symposium on Information Theory, ISIT 2016*, pages 1217–1221. IEEE, 2016.

[24] Michael I. Miller and Joseph A. O'Sullivan. Entropies and combinatorics of random branching processes and context-free languages. *IEEE Transactions Information Theory*, 38(4):1292–1310, 1992.

[25] Dimbinaina Ralaivaosaona and Stephan G. Wagner. Repeated fringe subtrees in random rooted trees. In *Proceedings of the Twelfth Workshop on Analytic Algorithmics and Combinatorics, ANALCO 2015*, pages 78–88. SIAM, 2015.

[26] Marcelo J. Weinberger, Jorma Rissanen, and Meir Feder. A universal finite memory source. *IEEE Transactions of Information Theory*, 41(3):643–652, 1995.

[27] En-Hui Yang and John C. Kieffer. Efficient universal lossless data compression algorithms based on a greedy sequential grammar transform - part one: Without context models. *IEEE Transactions on Information Theory*, 46(3):755–777, 2000.

[28] Jie Zhang, En-Hui Yang, and John C. Kieffer. A universal grammar-based code for lossless compression of binary trees. *IEEE Transactions on Information Theory*, 60(3):1373–1386, 2014.

[29] Jacob Ziv and Abraham Lempel. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, 24(5):530–536, 1978.

Moses Ganardi Moses Ganardi is a PhD student in Computer Science at the University of Siegen, Germany, under the supervision of Prof. Dr. Markus Lohrey. His main research interests are formal languages and automata theory, and their applications in the context of streaming algorithms. He received his B.Sc. degree in Computer Science and Mathematics, in 2011 and 2012, respectively, and his M.Sc.degree in Computer Science in 2013 from the RWTH Aachen, Germany.

Danny Hucke Danny Hucke is a PhD student under the supervision of Prof. Dr. Markus Lohrey at the University of Siegen, Germany. He works in the area of theoretical computer science and his main research interests are data compression, in particular grammar-based compression, and streaming algorithms. He received a diploma in Applied Mathematics from the HTWK Leipzig, Germany (2010) and a B.Sc. degree in Computer Science from the University of Leipzig, Germany (2013).

Markus Lohrey Markus Lohrey received his PhD from the University of Stuttgart, Germany, in 1999. From 2007 till 2013 he was professor for algebraic and logical foundations at University of Leipzig, Germany. Since 2013 he is professor for theoretical computer science at the University of Siegen, Germany. His working areas are algorithmic algebra, automata theory, and data compression. He published more than 150 papers in international journals and conferences.

Louisa Seelbach Benkner Louisa Seelbach Benkner is a PhD student in Computer Science under the supervision of Prof. Dr. Markus Lohrey at the University of Siegen, Germany. Her research centers on data compression, such as grammar-based compression of strings and trees. She received a B.Sc. degree in Mathematics in 2015 and a M.Sc. degree in Mathematics in 2017, both from the University of Siegen, Germany.