Deciding Sparseness of Regular Languages of Finite Trees and Infinite Words

Kord Eickmeyer and Georg Schindling
Technische Universität Darmstadt

July 2, 2025

Abstract

We study the notion of sparseness for regular languages over finite trees and infinite words. A language of trees is called sparse if the relative number of n-node trees in the language tends to zero, and a language of infinite words is called sparse if it has measure zero in the Bernoulli probability space. We show that sparseness is decidable for regular tree languages and for regular languages of infinite words. For trees, we provide characterisations in terms of forbidden subtrees and tree automata, leading to a linear time decision procedure. For infinite words, we present a characterisation via infix completeness and give a novel proof of decidability. Moreover, in the non-sparse case, our algorithm computes a measurable subset of accepted words that can serve as counterexamples in almost-sure model checking. Our findings have applications to automata based model checking in formal verifications and XML schemas, among others.

1 Introduction

We call a language of finite objects *sparse* if the relative number of n-element objects in the language tends to zero: For a language $L \subseteq \Sigma^*$ of words we set $\mathbb{P}_n(L) \coloneqq \frac{|L \cap \Sigma^n|}{|\Sigma^n|}$ and call L sparse if $\lim_{n\to\infty} \mathbb{P}_n(L) = 0$. Likewise, if L_T is a language of Σ-labelled trees and $\mathcal{B}_n^\Sigma$ is the set of all such trees with n nodes, we set $\mathbb{P}_n(L_T) \coloneqq \frac{|L_T \cap \mathcal{B}_n^\Sigma|}{|\mathcal{B}_n^\Sigma|}$ and call L_T sparse if $\lim_{n\to\infty} \mathbb{P}_n(L_T) = 0$. For a language $L_\omega \subseteq \Sigma^\omega$ of infinite words we turn Σ^ω into a Bernoulli probability space and call L_ω sparse if it has measure 0.

Regular languages play an important rôle both in applied and in theoretical computer science. Apart from appearing in practical applications such as pattern matching, one reason for their importance from a theoretical perspective is that they allow for several seemingly unrelated characterisations, in particular by finite monoids, finite automata, and as being definable in monadic second-order logic (MSO) by the Büchi-Elgot-Trakhtenbrot theorem [Bü60, Elg61, Tra61]. Many problems that are undecidable or at least computationally hard for general classes of languages are computable for regular languages. In particular the emptiness problem and, more generally, the subset relation between regular languages is efficiently decidable.

Naturally, the techniques used in dealing with regular languages have been adapted to other scenarios. Two of the most important and successful generalisations have been to languages of finite trees and languages of infinite words, also called ω-languages. In both these settings, the correspondence between regular languages, automata, and definability in MSO still holds (see [Don70, TW68] and [Bü90]), yielding similar algorithmic properties as for regular languages of finite words.

We provide a characterisation of sparse regular tree languages via *forbidden subtrees* and *tree automata* (Theorem 3.7). As a consequence, we deduce that sparseness of regular tree languages is decidable in linear time (Corollary 3.9). Furthermore, our results also provide a method for proving non-regularity of sparse tree languages (Corollary 3.11). Finally, we observe that our characterisation by forbidden subtrees does not hold for non-uniform models of random binary trees, specifically the random binary search tree model (Example 3.14).

A major motivation for regular ω-languages, on the other hand, is in model checking of reactive systems [BKL08]. In this setting, we are given a formal specification of a system (e.g. some piece of hardware or software) and a specification of properties that the system should have or not have. If both the system and the property are specified in linear temporal logic (LTL), model checking essentially reduces to checking whether a certain regular ω-language is empty or not. Theorem 4.4 extends this to *almost sure model checking*, i.e. the question whether a certain system *almost surely* satisfies a given

property. If the system fails to do so, it is even possible to compute a counterexample, in this case a set of ω-words of strictly positive measure that almost surely violates the condition to be checked.

Note that because the complement of a regular language is again regular, and L is sparse if and only if its complement has (asymptotic) probability 1, the same algorithms can be used to check whether a language of trees or ω-words has (asymptotic) probability 1.

Techniques and Related Work

Sparseness of regular languages was investigated by Sin'ya [Sin15]. He showed that a regular language $L \subseteq \Sigma^*$ over a finite alphabet Σ is sparse if, and only if, L has an excluded factor, i.e. $\Sigma^* x \Sigma^* \cap L = \emptyset$ for some $x \in \Sigma^*$. The proof of this characterisation was simplified by Koga in [Kog19]. We build on their insights and generalize them significantly to handle the richer settings of regular tree languages and ω-languages.

The trees we are most interested in are *rooted*, *ordered*, *unranked* trees, i.e. trees with a distinguished root node in which the number of children of a node is unbounded and for every node there is a linear order on the set of its children. The terms *siblinged* or *planar* are sometimes used for what we call ordered, e.g. in [FS09, BS09]. We state our results on regular tree languages in terms of binary trees with distinguished left and right children, cf. Section 2.2. This allows for a simpler definition of tree automata and comes at no loss of generality because ordered unranked trees are bijectively MSO-interpretable in binary trees, a folklore technique that is spelled out, for example, in [FG06, Sec. 10.2].

Because regular languages exactly capture the expressive power of monadic second-order logic (MSO), the (asymptotic) probability of regular languages can be rephrased as the (asymptotic) probability that a random structure satisfies a certain MSO sentence. This has been well studied: For every first-order logic (FO) sentence with relational signature, the probability that a random labelled structure satisfies the sentence converges to either 0 or 1 (this was proved independently in [GKLT69] and [Fag76], cf. [EF99, Ch. 4]), a result that has been extended by Spencer [Spe13] to Erdős-Rényi graphs with more general edge probabilities. The fact that every sentence is satisfied asymptotically with probability 0 or 1 is usually called 0-1 *law*. For monadic second-order logic (MSO), McColm [McC02] showed a 0-1 law for MSO on labelled trees. Note that McColm (crucially) considers trees without a distinguished root, while our trees are rooted. More recently, Malyshkin and Zhukovskii [MZ21] showed a 0-1 law for MSO on finite trees with probability measures other than uniform probabilities, namely uniform and preferential attachment. Further results on the limit behaviour of MSO sentences on various graph classes can be found in [DK19] and [AKNS18]. These results are not directly applicable to our setting. In fact, there can be no 0-1 law for regular languages: Both in trees and in ω-words there is some designated element (such as the first position in a word or the root of a tree), and the set of all structures in which this element gets a specific label is a regular set with (asymptotic) density strictly between 0 and 1.

Recently, Niwiński et al. [NPS23] proved that the exact probability of acceptance for MSO formulae in infinite random trees is computable. However, this can not be used to decide sparseness for languages of finite trees or infinite words.

A necessary and sufficient condition for an ω-language to have strictly positive measure has already been proved by Staiger [Sta98]. However, Staiger does not address the question of decidability of this property, and his proof uses deep results from topology and Kolmogorov complexity, namely a connection between the *subword complexity* of an ω-word ξ and the *Hausdorff dimension* of the regular ω-languages containing ξ. Courcoubetis et al. [CY95] treated the sparseness problem for ω-languages under the name *probabilistic emptyness* and showed it to be decidable with methods similar to ours. By relating non-sparseness to infix completeness, our (shorter) proof has the additional feature of giving an exact characterisation of non-sparse ω-languages. Furthermore, the algorithm we present in Thm. 4.4 is the first to provide a concise representation of a set M of ω-words that is almost a subset of L, in the sense that $\mathbb{P}(M \setminus L) = 0$. This is extremely valuable in the context of LTL model checking, as it amounts to a non-negligible set of counterexamples.

2 Preliminaries

2.1 Languages and Automata

By Σ we denote a finite alphabet assuming $|\Sigma| > 1$ to avoid trivialities. For $k \geq 0$ we denote by Σ^*, Σ^k, and $\Sigma^{\leq k}$ the sets of all strings, those of length k, and of length at most k, respectively. The empty string

is denoted by ε. A *language* is any subset $L \subseteq \Sigma^*$. Concatenation of strings $u, v \in \Sigma^*$ is denoted by $u \cdot v$ or just uv, and for a language L we set $L^* \coloneqq \{u_1 \cdots u_k \mid k \geq 0, u_1, \ldots, u_k \in L\}$.

An *ω-word* over the alphabet Σ is an infinite sequence $a_0 a_1 a_2 \cdots$ of letters $a_i \in \Sigma$. The set of all ω-words is denoted by Σ^ω, and an ω-language is any subset $L \subseteq \Sigma^\omega$. A (finite) word $u \in \Sigma^*$ and an ω-word $v \in \Sigma^\omega$ may be concatenated to $uv \in \Sigma^\omega$, and for a language $U \subseteq \Sigma^*$ and an ω-language $V \subseteq \Sigma^\omega$ we set

$$U^\omega \coloneqq \{u_1 u_2 u_3 \cdots \mid u_1, u_2, \ldots \in U\} \qquad \text{and} \qquad UV \coloneqq \{uv \mid u \in U \text{ and } v \in V\}.$$

A word $u \in \Sigma^*$ is a *prefix* of $v \in \Sigma^* \cup \Sigma^\omega$, written $u \preceq v$, if $v = uw$ for some $w \in \Sigma^* \cup \Sigma^\omega$.

A *deterministic finite automaton (DFA)* over the alphabet Σ is a tuple $\mathcal{A} = (\Sigma, Q, q_0, \delta, A)$ consisting of a finite set Q of *states*, a designated *initial state* $q_0 \in Q$, a *transition function* $\delta \colon Q \times \Sigma \to Q$ and a set $A \subseteq Q$ of *accepting states*. We extend δ to a function $\hat{\delta} \colon Q \times \Sigma^* \to Q$ by setting $\hat{\delta}(q, \varepsilon) \coloneqq q$ and $\hat{\delta}(q, aw) \coloneqq \hat{\delta}\big(\delta(q, a), w\big)$ for $a \in \Sigma$ and $w \in \Sigma^*$. The automaton $\mathcal{A}$ *accepts* a word $w \in \Sigma^*$ if $\hat{\delta}(q_0, w) \in A$. The *language accepted by* $\mathcal{A}$ is the set of words accepted by it. A language is called *regular* if it is accepted by some DFA.

For states $q, q' \in Q$ we say that q' is *reachable* from q, written $q \rightsquigarrow q'$, if $\hat{\delta}(q, w) = q'$ for some word $w \in \Sigma^*$. If q' is reachable from the initial state q_0 we just call q' reachable. The relation $\rightsquigarrow$ is obviously reflexive and transitive, so the relation $q \approx q' \Leftrightarrow (q \rightsquigarrow q' \text{ and } q' \rightsquigarrow q)$ is an equivalence relation. Its equivalence classes are called the *reachability classes* of $\mathcal{A}$. A reachability class C is called *closed* if $\delta(q, a) \in C$ for all $q \in C$ and $a \in \Sigma$

There are various generalisations of finite automata to ω-words, resulting in a robust concept of regular ω-languages, cf. [Tho97]. We use the following characterisation:

Theorem 2.1 (cf. [Sta97], Thm. 3.2)**.** *An ω-language $L \subseteq \Sigma^\omega$ is regular if and only if there is a $k \geq 1$ and regular languages $U_i, V_i \subseteq \Sigma^*$ for $i = 1, \ldots, k$ such that $L = \bigcup_{i=1}^k U_i V_i^\omega$.*

2.2 Tree Languages

We will mostly be concerned with *binary trees* in the sense of Knuth [Knu97, 2.3]. These trees have a distinguished root node, and every node may have a left and/or a right child. We formalise this as a prefix-closed language $T \subseteq D^*$ over the alphabet $D = \{l, r\}$. The empty word ε denotes the root of the tree T, a node $u \in T$ may have children $ul, ur \in T$ and is the *parent* of these.

For a finite alphabet Σ and binary tree T we call a function $\lambda \colon T \to \Sigma$ a *Σ-labelling* of T and the pair (T, λ) a Σ-labelled binary tree. We denote the set of all finite Σ-labelled binary trees by $\mathcal{B}^\Sigma$. We often just write $T \in \mathcal{B}^\Sigma$ and refer to the labelling as λ_T when necessary. A set $L \subseteq \mathcal{B}^\Sigma$ is called a *tree language*. Note that $\mathcal{B}^\Sigma$ contains the empty tree $\emptyset$.

For $S, T \in \mathcal{B}^\Sigma$ we say that S is a *subtree* of T and write $S \preceq T$ if there exists $u \in D^*$ such that $uD^* \cap T = uS$ and $\lambda_S(v) = \lambda_T(uv)$ for all nodes $v \in S$. Each node $u \in T$ induces a subtree $T(u)$ of T consisting of u and all its descendants. We write T_l for the *left subtree* $T(l)$ and T_r for the *right subtree* $T(r)$ of the root of T, respectively.

For two trees $S, T \in \mathcal{B}^\Sigma$ and $a \in \Sigma$ we define the Σ-labelled binary tree $\mathrm{conc}_a(S, T)$ which consists of the root labelled with a and has S and T as left and right subtrees respectively.

For a tree language $L \subseteq \mathcal{B}^\Sigma$ and $T \in \mathcal{B}^\Sigma$ we define the tree languages

$$LT \coloneqq \bigcup_{a \in \Sigma} \{\mathrm{conc}_a(S, T) \mid S \in L\}$$
$$LT^{-1} \coloneqq \{S \in \mathcal{B}^\Sigma \mid \text{ There exists } a \in \Sigma \text{ with } \mathrm{conc}_a(S, T) \in L\}$$

and $TL, T^{-1}L$ are defined analogously. For trees $T_1, \ldots, T_k \in \mathcal{B}^\Sigma$ we inductively set

$$L[T_1]^{-1} \coloneqq LT_1^{-1} \cup T_1^{-1}L$$
$$L[T_k, \ldots, T_1]^{-1} \coloneqq (L[T_{k-1}, \ldots, T_1]^{-1})T_k^{-1} \cup T_k^{-1}(L[T_{k-1}, \ldots, T_1]^{-1}),$$

so $L[T_k, \ldots, T_1]^{-1}$ is the language of all trees that can be concatenated successively with $T_k, \ldots, T_1$ to obtain a tree from L.

A *tree automaton* is a tuple $\mathcal{A} = (\Sigma, Q, \Delta, A)$ consisting of a finite set Q of *states*, a finite alphabet Σ, a *transition relation* $\Delta \subseteq (Q \cup \{\bot\}) \times (Q \cup \{\bot\}) \times \Sigma \times Q$, and a set $A \subseteq Q$ of *accepting* states. Given a Σ-labelled binary tree T, a *run* of $\mathcal{A}$ on T is a function $d \colon D^* \to (Q \cup \{\bot\})$ such that

- if $u \notin T$ then $d(u) = \bot$, and

- if $u \in T$ then $(d(ul), d(ur), \lambda_T(u), d(u)) \in \Delta$.

A run d is called *accepting* if $d(\varepsilon) \in A$ and we say that an automaton $\mathcal{A}$ *accepts* a tree T if there is an accepting run of $\mathcal{A}$ on T.

For states $q \in Q$ and $q' \in Q \cup \{\bot\}$, we say that q' is 1-step reachable from q (written $q \rightsquigarrow_1 q'$) if $(q, \tilde{q}, a, q') \in \Delta$ or $(\tilde{q}, q, a, q') \in \Delta$ for some $a \in \Sigma$ and $\tilde{q} \in Q \cup \{\bot\}$. The reflexive transitive closure $\rightsquigarrow := \rightsquigarrow_1^*$ is called reachability, and a state $q \in Q$ is called *reachable* if $\bot \rightsquigarrow q$. Equivalently, q is reachable if and only if there exists a tree $T \in \mathcal{B}^\Sigma$ and a run d of $\mathcal{A}$ on T with $d(\varepsilon) = q$. We call $\mathcal{A}$ *reduced* if every state is reachable.

2.3 Random Trees and ω-words

For $n \in \mathbb{N}$ we denote by $\mathcal{B}_n^\Sigma$ and $\mathcal{B}_{<n}^\Sigma$ the set of Σ-labelled binary trees of size n and size strictly less than n respectively. We consider $\mathcal{B}_n^\Sigma$ as a finite discrete probability space equipped with the uniform distribution $\mathbb{P}_n$ by setting $\mathbb{P}_n(T) = \frac{1}{|\mathcal{B}_n^\Sigma|}$ for $T \in \mathcal{B}_n^\Sigma$. Note that $|\mathcal{B}_n^\Sigma| = C_n \cdot |\Sigma|^n$, where $C_n = \frac{1}{n+1}\binom{2n}{n} \approx \frac{4^n}{n\sqrt{\pi n}}$ is the n-th *Catalan number*, cf. [Knu97, 2.3.4.4].

The *asymptotic density* (or *asymptotic probability*) of a tree language $L \subseteq \mathcal{B}^\Sigma$ is defined as

$$\mathbb{P}_{\lim}(L) := \lim_{n\to\infty} \mathbb{P}_n(L \cap \mathcal{B}_n^\Sigma) = \lim_{n\to\infty} \frac{|L \cap \mathcal{B}_n^\Sigma|}{C_n \cdot |\Sigma|^n},$$

given the limit exists. We set $\overline{\mathbb{P}}_{\lim}(L) := \limsup_{n\to\infty} \mathbb{P}_n(L \cap \mathcal{B}_n^\Sigma)$ and obtain the following lemma, which is easily verified:

Lemma 2.2. *For all $L_1, L_2 \subseteq \mathcal{B}^\Sigma$, the following hold:*

1. $\overline{\mathbb{P}}_{\lim}(L) = 0 \Leftrightarrow \mathbb{P}_{\lim}(L) = 0$
2. $L_1 \subseteq L_2 \Rightarrow \overline{\mathbb{P}}_{\lim}(L_1) \leq \overline{\mathbb{P}}_{\lim}(L_2)$
3. $\overline{\mathbb{P}}_{\lim}(L_1 \cup L_2) \leq \overline{\mathbb{P}}_{\lim}(L_1) + \overline{\mathbb{P}}_{\lim}(L_2)$
4. $\overline{\mathbb{P}}_{\lim}(\mathcal{B}^\Sigma) = 1$.

We use standard terminology from probability theory, cf. [Wil91]. We turn the set Σ^ω of ω-words over the finite alphabet Σ into a probability space by making each of the projections $\pi_i \colon \Sigma^\omega \to \Sigma, w_1 w_2 \ldots \mapsto w_i$ measurable. By $\mathbb{P}$ we denote the probability measure for which the projections are iid random variables with $\mathbb{P}(w_i = a) = |\Sigma|^{-1}$ for every $i \geq 1$ and $a \in \Sigma$. Note that for any $U, V \subseteq \Sigma^*$, $UV^\omega = \bigcap_{k\geq 1} \bigcup_{\ell \geq k} \{w_1 w_2 \ldots \mid w_1 \ldots w_\ell \in UV^*\}$ is measurable, so by Thm. 2.1, every ω-regular $L \subseteq \Sigma^\omega$ is measurable and the probability (or measure) $\mathbb{P}(L)$ is well-defined.

We review some basic facts about discrete-time Markov chains with a finite state space, cf. [Nor97]: Fix a finite set I of *states* and for every $i, j \in I$ a *transition probability* $p_{ij} \geq 0$ such that $\sum_{j\in I} p_{ij} = 1$ for every $i \in I$. A *Markov chain* with state space I and transition probabilities $P = (p_{ij})_{i,j\in I}$ is a sequence $(X_t)_{t\in\mathbb{N}}$ of random variables taking values in I such that $\mathbb{P}(X_{t+1} = j \mid X_t = i) = p_{ij}$ for every $t \geq 0$ and $i, j \in I$. The probability distribution of X_0 is called *initial distribution* of the chain. The initial distribution and the transition probabilities P together determine the joint distributions of the X_t by

$$\mathbb{P}(X_0 = i_0, \ldots, X_t = i_t) = \mathbb{P}(X_0 = i_0) \cdot p_{i_0 i_1} \cdots p_{i_{t-1} i_t}.$$

If $\mathbb{P}(X_0 = j) = \delta_{ij}$ we say that the chain is started in state i and denote the resulting probability distribution by $\mathbb{P}_i$. With this definition, $\mathbb{P}(X_{s+t} = j \mid X_s = i) = \mathbb{P}_i(X_t = j)$. A state $i \in I$ is called *recurrent* if $\mathbb{P}_i(X_t = i \text{ for infinitely many t}) = 1$. We say that a state $i \in I$ *leads to* a state $j \in I$, written $i \to j$, if $\mathbb{P}_i(X_t = j \text{ for some } t \in \mathbb{N}) > 0$, and that i and j *communicate* (written $i \leftrightarrow j$) if both $i \to j$ and $j \to i$. Then $\leftrightarrow$ is an equivalence relation on I and its equivalence classes are just called *classes* of states. A class $C \subseteq I$ is called *closed* if $i \in C$ and $i \to j$ imply $j \in C$. We need the following theorem:

Theorem 2.3 (cf. [Nor97, Thm. 1.5.6])**.** *If $C \subseteq I$ is a closed class, every $i \in C$ is recurrent.*

Corollary 2.4. *If $C \subseteq I$ is a closed class and $i \in C$, then $\mathbb{P}(X_t = i \text{ infinitely often} | X_t \in C \text{ for some } t) = 1$ for every $s \in \mathbb{N}$.*

3 Characterising Sparseness of Regular Tree Languages

In this section we exactly characterise regular tree languages with asymptotic density 0 by excluded factors, namely *forbidden subtrees*. This generalises Sin'ya's result for regular languages. The well-known *infinite monkey theorem* states that a language of finite words $L \subseteq \Sigma^*$ has asymptotic density 1 if $\Sigma^* x \Sigma^* \subseteq L$ for some $x \in \Sigma^*$. This has been generalised to tree languages by Asada et al. [AKST19, Thm. 2.13], who prove that *contexts* of up to logarithmic size appear asymptotically almost surely in certain regular tree languages. We only need a weaker version stated in Theorem 3.1, and give a comparatively short proof of it using methods from *analytic combinatorics* [FS09]. In Theorem 3.7 we then show that, for a *regular* tree language, the existence of a forbidden subtree is a necessary condition for sparseness.

Theorem 3.1. $\mathbb{P}_{\lim}(\{T \in \mathcal{B}^\Sigma \mid S \preceq T\}) = 1$ *for every nonempty* $S \in \mathcal{B}^\Sigma$.

Proof. First, note that

$$\begin{aligned}\overline{\mathbb{P}}_{\lim}(\{T \in \mathcal{B}^\Sigma \mid S \preceq T\}) &= \limsup(1 - \mathbb{P}\{T \in \mathcal{B}^\Sigma \mid S \not\preceq T\}) \\ &= 1 - \liminf_{n\to\infty} \frac{|\{T \in \mathcal{B}^\Sigma_n \mid S \not\preceq T\}|}{C_n|\Sigma|^n}.\end{aligned}$$

We fix a nonempty tree $S \in \mathcal{B}^\Sigma$ and examine the asymptotic behaviour of the sequence

$$a_n \coloneqq |\{T \in \mathcal{B}^\Sigma_n \mid S \not\preceq T\}|$$

following the approach of [FS09, Example III.41]. To analyse the generating function of the sequence $(a_n)_n$, we denote by $f_{n,k}$ the number of Σ-labelled binary trees of size n that contain S as a subtree at k different positions, and let

$$f(u,z) \coloneqq \sum_{n,k\geq 0} f_{n,k} z^n u^k$$

be its bivariate generating function. In particular $a_n = f_{n,0}$ and $f(0,z) = \sum_n a_n z^n$ is the generating function of the sequence $(a_n)_{n\geq 0}$.

By $\omega(T)$ we denote the number of distinct occurrences of S in a tree $T \in \mathcal{B}^\Sigma$. Then $\omega(\emptyset) = 0$ and since S can occur in the left or right subtree, or be the whole tree T, we get $\omega(T) = \omega(T_l) + \omega(T_r) + [T = S]$, where $[T = S]$ is 1 if $T = S$ and 0 otherwise, for $T \neq \emptyset$. For the function $u \mapsto u^{\omega(T)}$ this can be rewritten as

$$u^{\omega(T)} = u^{\omega(T_l)} u^{\omega(T_r)} u^{[T=S]} = u^{\omega(T_l)} u^{\omega(T_r)} + [T = S](u-1), \tag{1}$$

and justifying algebraic manipulations of formal power series as in [FS09, A.5.], we get:

$$\begin{aligned} f(u,z) &= \sum_{n=0}^\infty z^n \sum_{k=0}^\infty u^k f_{n,k} = \sum_{n=0}^\infty z^n \sum_{T\in\mathcal{B}^\Sigma_n} u^{\omega(T)} \\ &\overset{(1)}{=} \sum_{n=0}^\infty z^n \sum_{T\in\mathcal{B}^\Sigma_n} \Big([T = S](u-1) + u^{\omega(T_l)} u^{\omega(T_r)}\Big) \\ &= \sum_{n=0}^\infty z^n \sum_{T\in\mathcal{B}^\Sigma_n} [T = S](u-1) + \sum_{n=0}^\infty z^n \sum_{T\in\mathcal{B}^\Sigma_n} u^{\omega(T_l)} u^{\omega(T_r)} \\ &= z^m(u-1) + 1 + \sum_{n=1}^\infty z^n \sum_{T\in\mathcal{B}^\Sigma_n} u^{\omega(T_l)} u^{\omega(T_r)} \end{aligned}$$

for $m \coloneqq |S|$. We set $f_n(u) \coloneqq \sum_{T \in \mathcal{B}_n^\Sigma} u^{\omega(T)} = \sum_{k=0}^{\infty} u^k f_{n,k}$ and get

$$\begin{aligned} f(u,z) - z^m(u-1) - 1 &= \sum_{n=1}^{\infty} z^n \sum_{T \in \mathcal{B}_n^\Sigma} u^{\omega(T_l)} u^{\omega(T_r)} \\ &= \sum_{n=1}^{\infty} z^n \sum_{j=0}^{n-1} |\Sigma| \Big(\sum_{T_l \in \mathcal{B}_j^\Sigma} u^{\omega(T_l)} \Big) \Big(\sum_{T_r \in \mathcal{B}_{n-1-j}^\Sigma} u^{\omega(T_r)} \Big) \\ &= z|\Sigma| \sum_{n=1}^{\infty} z^{n-1} \sum_{j=0}^{n-1} \Big(\sum_{T_l \in \mathcal{B}_j^\Sigma} u^{\omega(T_l)} \Big) \Big(\sum_{T_r \in \mathcal{B}_{n-1-j}^\Sigma} u^{\omega(T_r)} \Big) \\ &= z|\Sigma| \sum_{n=0}^{\infty} \sum_{j=0}^{n} z^j f_j(u) \cdot z^{n-j} f_{n-j}(u) = z|\Sigma| f(u,z)^2. \end{aligned}$$

Solving this quadratic equation for $f(u,z)$ gives two candidate solutions

$$f(u,z) = \frac{1 \pm \sqrt{1 - 4z|\Sigma| - 4|\Sigma| z^{m+1}(u-1)}}{2z|\Sigma|},$$

and since $f(1, \frac{z}{|\Sigma|})$ is the generating function of the Catalan numbers, subtracting the square root gives the right solution. The generating function $f(0,z)$ of the sequence $(a_n)_{n \geq 0}$ is now given by

$$f(0,z) = \frac{1 - \sqrt{1 - 4z|\Sigma| + 4|\Sigma| z^{m+1}}}{2z|\Sigma|}.$$

The function $f(0,z)$ is analytic at 0 by extending it to $f(0,0) = 1$. The radius around 0, where $f(0,z)$ is analytic is exactly the radius R, for which the polynomial $p(z) \coloneqq 1 - 4|\Sigma|z + 4|\Sigma|z^{m+1}$ is non-zero (cf. analyticity of $\sqrt{1-z}$). Considering the reciprocal polynomial of p yields $R > \frac{1}{4|\Sigma|}$.

Finally, by [FS09, Theorem IV.7 (Exponential Growth Formula)] there exist a subexponential factor $(\eta_n)_n$ such that $a_n = R^{-n}\eta_n$. Also, by Stirling's formula there exists a subexponential factor $(\theta_n)_n$ such that $C_n = 4^{-n}\theta_n$. In total, this yields

$$\frac{|\{T \in \mathcal{B}_n^\Sigma \mid S \not\leq T\}|}{|\mathcal{B}_n^\Sigma|} = \frac{a_n}{C_n |\Sigma|^n} = \frac{R^{-n}\eta_n}{4^n \theta_n |\Sigma|^n} = \Big(\frac{1}{4R|\Sigma|} \Big)^n \frac{\eta_n}{\theta_n}.$$

Since the factors η_n and θ_n are subexponential the sequence converges to 0 as n tends to infinity. □

In order to show the converse for *regular* tree languages, we lift the proof from [Kog19] to the case of tree languages. First, we derive bounds for the asymptotic density of specific tree languages.

Lemma 3.2. *For $L \subseteq \mathcal{B}^\Sigma$ and $T, T_1, \ldots, T_k \in \mathcal{B}^\Sigma$ the following hold:*

1. $\overline{\mathbb{P}}_{\lim}(LT) = \frac{1}{|\Sigma|^{|T|} 4^{|T|+1}} \overline{\mathbb{P}}_{\lim}(L)$
2. $\overline{\mathbb{P}}_{\lim}(LT^{-1}), \overline{\mathbb{P}}_{\lim}(T^{-1}L) \leq |\Sigma|^{|T|} 4^{|T|+1} \overline{\mathbb{P}}_{\lim}(L)$
3. $\overline{\mathbb{P}}_{\lim}(L[T_k, \ldots, T_1]^{-1}) \leq 2^k |\Sigma|^{\sum_i |T_i|} 4^{\sum_i |T_i| + k} \overline{\mathbb{P}}_{\lim}(L)$

Proof. By

$$\begin{aligned} \mathbb{P}_n(LT \cap \mathcal{B}_n^\Sigma) &= \frac{|(L \cap \mathcal{B}_{n-|T|-1}^\Sigma)T|}{|\mathcal{B}_n^\Sigma|} = \frac{|L \cap \mathcal{B}_{n-|T|-1}^\Sigma|}{|\mathcal{B}_{n-|T|-1}^\Sigma|} \frac{|\Sigma| \cdot |\mathcal{B}_{n-|T|-1}^\Sigma|}{|\mathcal{B}_n^\Sigma|} \\ &= \mathbb{P}_{n-|T|-1}(L \cap \mathcal{B}_{n-|T|-1}^\Sigma) \frac{C_{n-|T|-1} |\Sigma|^{n-|T|}}{C_n |\Sigma|^n} \\ &= \mathbb{P}_{n-|T|-1}(L \cap \mathcal{B}_{n-|T|-1}^\Sigma) \frac{1}{|\Sigma|^{|T|}} \frac{C_{n-|T|-1}}{C_n}, \end{aligned}$$

taking the limes superior on both sides together with the identity $\lim_{n\to\infty} \frac{C_{n-k}}{C_n} = \frac{1}{4^k}$ proves the first part. For the second part note that $(LT^{-1})T \subseteq L$ and hence by Lemma 2.2 together with the first part we obtain

$$\overline{\mathbb{P}}_{\lim}(L) \geq \overline{\mathbb{P}}_{\lim}\big((LT^{-1})T\big) = \frac{1}{|\Sigma|^{|T|}4^{|T|+1}}\overline{\mathbb{P}}_{\lim}(LT^{-1}),$$

and likewise for $\overline{\mathbb{P}}_{\lim}(T^{-1}L)$. Finally, for the third part we have

$$\begin{aligned}
&\overline{\mathbb{P}}_{\lim}(L[T_k,\dots,T_1]^{-1}) \\
&= \overline{\mathbb{P}}_{\lim}\Big((L[T_{k-1},\dots,T_1]^{-1})T_k^{-1} \cup T_k^{-1}(L[T_{k-1},\dots,T_1]^{-1})\Big) \\
&\leq \overline{\mathbb{P}}_{\lim}\Big((L[T_{k-1},\dots,T_1]^{-1})T_k^{-1}\Big) + \overline{\mathbb{P}}_{\lim}\Big((L[T_{k-1},\dots,T_1]^{-1})T_k^{-1}\Big) \\
&\leq 2|\Sigma|^{|T_k|}4^{|T_k|+1}\overline{\mathbb{P}}_{\lim}(L[T_{k-1},\dots,T_1]^{-1})
\end{aligned}$$

by using the second part for the last inequality. Hence, the claim follows by induction. □

Lemma 3.3. *Let $\mathcal{A} = (Q, \Sigma, \Delta, A)$ be a tree automaton. For every reachable state $q \in Q$ there exists a tree $T \in \mathcal{B}^\Sigma$ with $|T| \leq 2^{|Q|} - 1$ such that a run of $\mathcal{A}$ on T ends in q.*

Proof. Let $q \in Q$ be reachable by $\mathcal{A}$, so there exists $T \in \mathcal{B}^\Sigma$ such that a run of $\mathcal{A}$ on T ends in q. If $|T| \leq 2^{|Q|} - 1$ we are done, so assume $|T| > 2^{|Q|} - 1$. Then there exists $w \in T$ with $|w| \geq |Q|$. Let $d\colon D^* \to Q \cup \{\bot\}$ be a run of $\mathcal{A}$ on T, so for all $u \in T$ it holds $(d(ul), d(ur), \lambda(u), d(u)) \in \Delta$. We obtain a sequence $[d(p^n(w))]_{n=0}^{|w|}$ of states, which $\mathcal{A}$ passes from w to $p^{|w|}(w) = \varepsilon$. Since $|w| \geq |Q|$, by the pigeonhole principle, there must be a state in Q which occurs twice in $[d(p^n(w))]_{n=0}^{|w|}$. Let $i, j \in \{0,\dots,|w|\}$ with $i \neq j$ and $d(p^i(w)) = d(p^j(w))$, then either $T(p^i(w)) \preceq T(p^j(w))$ or $T(p^j(w)) \preceq T(p^i(w))$. We assume $T(p^j(w)) \preceq T(p^i(w))$, so $j > i$ (the other case is analogous). We obtain a new tree T by replacing $T(p^i(w))$ by $T(p^j(w))$ in T and set $w^1 \coloneqq w_0 \dots w_{i-1} w_j \dots w_{|w|+1}$, which is the new node at the position of w. Then $|w^1| = |w| - |i-j| < |w|$. Since $\mathcal{A}$ ends in the same state after running on $T(p^j(w))$ and $T(p^i(w))$, it still ends in q after running on T_1. If $|w^1| \geq |Q|$ the same argument applies for the sequence $[d(w), d(p(w)), \dots, d(p^j(w)), d(p^{i+1}(w)), \dots, d(\varepsilon)]$ and we iteratively obtain $|w^\ell| < |Q|$ after at most ℓ iterations. The same argument can be applied to every node $v \in T_\ell$ with $|v| \geq |Q|$ until for all nodes v in the resulting tree it holds $|v| < |Q|$. A binary tree with this property has depth at most $|Q| - 1$, so its size is bounded by $\sum_{k=0}^{|Q|-1} 2^k = 2^{|Q|} - 1$. □

Next, we show that if a binary tree S occurs as a subtree in a regular language L, then there is a tree $T \in L$ with $S \preceq T$ that is 'not much larger' than S:

Lemma 3.4. *For every regular tree language $L \subseteq \mathcal{B}^\Sigma$ there exists $n \in \mathbb{N}$ such that for all $S \in \mathcal{B}^\Sigma$ and $T \in L$ with $S \preceq T$ there exist $T_1, \dots, T_k \in \mathcal{B}^\Sigma_{<2^n}$ with $k \leq n$ such that $S \in L[T_k, \dots, T_1]^{-1}$.*

Proof. Let $\mathcal{A} = (\Sigma, Q, \Delta, A)$ be a tree automaton recognising the language L. For $T \in L$ and $S \in \mathcal{B}^\Sigma$ with $S \preceq T$ there exists $w = w_1 \cdots w_\ell \in T$ with $S = T(w)$. Let $d\colon D^* \to Q \cup \{\bot\}$ be a run of $\mathcal{A}$ on T and $d_i = d(w_1 \cdots w_i) \in Q$ be state of $\mathcal{A}$ at node $w_1 \cdots w_i$ in this run, for $i = 0, \dots, \ell$. Then $d_0 \in A$ (because $\mathcal{A}$ accepts T) and $d_i \rightsquigarrow_1 d_{i-1}$ for $i = 1, \dots, \ell$. If $d_i = d_j$ for some $i < j$ we remove that subsequence $d_i, \dots, d_{j-1}$ and repeat this process until we get a sequence $d'_0, \dots, d'_m$ with $m < |Q|$.

Note that $d_S : D^* \to Q \cup \{\bot\}$ with $d_S(v) \coloneqq d(wv)$ is a run of $\mathcal{A}$ on S with $d_S(\epsilon) = d(w) = d_\ell = d'_n$, and our reduced sequence d' still satisfies $d'_i \rightsquigarrow_1 d'_{i-1}$ for $i = 1, \dots, m$. The result, with $n = |Q|$, now follows from Lemma 3.3 and the definition of $\rightsquigarrow_1$. □

We use the previous lemma to show that if a regular tree language L is sparse, then it must already admit a *forbidden subtree*. That is, a fixed tree S which does not occur as subtree of any tree in L.

Theorem 3.5. *Let $L \subseteq \mathcal{B}^\Sigma$ be a regular tree language with $\mathbb{P}_{\lim}(L) = 0$. Then there exists $S \in \mathcal{B}^\Sigma$ such that $\{T \in \mathcal{B}^\Sigma \mid S \preceq T\} \cap L = \emptyset$.*

Proof. We argue by contraposition and assume that for all $S \in \mathcal{B}^\Sigma$ we have $\{T \in \mathcal{B}^\Sigma \mid S \preceq T\} \cap L \neq \emptyset$. That is, for every $S \in \mathcal{B}^\Sigma$ there exists $T \in L$ with $S \preceq T$. By Lemma 3.4, we infer that for every $S \in \mathcal{B}^\Sigma$ there exist $k \leq n$ and $T_1, \dots, T_k \in \mathcal{B}^\Sigma_{<2^n}$ such that $S \in L[T_k, \dots, T_1]^{-1}$. This in turn is equivalent to

$$\mathcal{B}^\Sigma \subseteq \bigcup_{k=1}^{n} \bigcup_{T_1,\dots,T_k \in \mathcal{B}^\Sigma_{<2^n}} L[T_k, \dots, T_1]^{-1}.$$

Using Lemma 2.2 and k successive applications of Lemma 3.2 we obtain:

$$\begin{aligned} 1 = \overline{\mathbb{P}}_{\lim}(\mathcal{B}^\Sigma) &\leq \overline{\mathbb{P}}_{\lim}\Big(\bigcup_{k=1}^{n} \bigcup_{T_1,\dots,T_k \in \mathcal{B}^\Sigma_{<2^n}} L[T_k,\dots,T_1]^{-1}\Big) \\ &\leq \sum_{k=1}^{n} \sum_{T_1,\dots,T_k \in \mathcal{B}^\Sigma_{<2^n}} \overline{\mathbb{P}}_{\lim}(L[T_k,\dots,T_1]^{-1}) \\ &\leq \sum_{k=1}^{n} \sum_{T_1,\dots,T_k \in \mathcal{B}^\Sigma_{<2^n}} 2^k \prod_{j=1}^{k} |\Sigma|^{|T_j|} 4^{|T_j|+1} \overline{\mathbb{P}}_{\lim}(L) \end{aligned}$$

Thus, we conclude that $\mathbb{P}_{\lim}(L) > 0$. □

This converse of the infinite monkey theorem for regular tree languages provides a characterisation of sparseness in terms of *forbidden subtrees*. In order to also obtain such a characterisation in terms of tree automata akin to [Sin15], we give the following definition.

Definition 3.6. Let $\mathcal{A} = (\Sigma, Q, \Delta, A)$ be a tree automaton. A set of states $V \subseteq Q$ is called a *sink* if for all $q \in V$ and all $q_l, q_r \in Q \cup \{\bot\}, a \in \Sigma$ it holds that $\delta(q_l, q, a), \delta(q, q_r, a) \subseteq V$. That is, V is a sink exactly if every run of $\mathcal{A}$ remains in V once it entered a state in V.

Finally, we conclude the following characterisation of sparse regular tree languages.

Theorem 3.7. *Let L be a regular tree language and $\mathcal{A} = (\Sigma, Q, \Delta, A)$ be a reduced tree automaton recognising L. Then the following assertions are equivalent:*

1. $\mathbb{P}_{\lim}(L) = 0$
2. *There exists a tree $S \in \mathcal{B}^\Sigma$ such that $\{T \in \mathcal{B}^\Sigma \mid S \preceq T\} \subseteq \mathcal{B}^\Sigma \setminus L$*
3. *$\mathcal{A}$ has a sink $V \subseteq Q$ with $V \cap A = \emptyset$*

Proof. Theorems 3.1 and 3.5 together imply 1 ⇔ 2.
2 ⇒ 3: Let $S \in \mathcal{B}^\Sigma$ such that $\{T \in \mathcal{B}^\Sigma \mid S \preceq T\} \subseteq \mathcal{B}^\Sigma \setminus L$. Let $q_0 \in Q$ be a state in which $\mathcal{A}$ ends after reading S. Then $q_0 \in Q \setminus A$ since $\mathcal{A}$ has to reject S because all trees which contain S as subtree have to be rejected. Therefore, all $q_l, q_r \in Q \cup \{\bot\}$ and $a \in \Sigma$ satisfy $\delta(q_l, q_0, a) \subseteq Q \setminus A$ and $\delta(q_0, q_r, a) \subseteq Q \setminus A$ because otherwise one could construct a tree (since every state is reachable) which contains S as subtree and is accepted by $\mathcal{A}$. Hence, there exists a sink $S \subseteq Q \setminus A$ which contains q_0.
3 ⇒ 2: Let $V \subseteq Q \setminus A$ be a sink of $\mathcal{A}$. There exists a tree S on which $\mathcal{A}$ ends in a state of V. If $\mathcal{A}$ runs on any tree containing S as a subtree, $\mathcal{A}$ cannot leave the sink and thus cannot leave $Q \setminus A$. This yields $\{T \in \mathcal{B}^\Sigma \mid S \preceq T\} \subseteq L$. □

Remark 3.8. By duality, Theorem 3.7 also provides a characterisation of regular tree languages L with asymptotic density 1: It holds $\mathbb{P}_{\lim}(L) = 1$ if and only if there exists a tree S such that $S \preceq T$ implies $T \in L$, and this is the case if and only if an automaton recognising L has a sink consisting of accepting states.

As a consequence of this characterisation, we obtain a simple linear time algorithm for deciding sparseness (or denseness) of regular tree languages akin to the algorithm in [Sin15]. Note that in contrast to the characterisation given there, here we do not need to require that a sink is a strongly connected component. If there is a sink $V \subseteq Q$ with $V \subseteq Q \setminus A$, there cannot be a sink $V' \subseteq A$.

Corollary 3.9. *Let $L \subseteq \mathcal{B}^\Sigma$ be a regular tree language. There is an algorithm deciding whether L has asymptotic density 0 or 1 in time $O(n)$, where n is the number of states of a given deterministic tree automaton $\mathcal{A}$ recognising L.*

Proof. We define the *state graph* of a deterministic tree automaton $\mathcal{A} = (Q, \Sigma, \delta, A)$ as $G_\mathcal{A} = (Q, \{(q, q') \mid \exists s \in Q, a \in \Sigma.\, \delta(s, q, a) = q' \text{ or } \delta(q, s, a) = q'\})$. For a set $V \subseteq Q$ we define $N^+_{G_\mathcal{A}}(V) := \{q' \in Q \setminus V : \exists q \in V.(q, q') \in E(G_\mathcal{A})\}$. A *strongly connected component* of $\mathcal{A}$ is a strongly connected component of the directed graph $G_\mathcal{A}$. The linear time algorithm now is as follows:

1. Compute the set of strongly connected components of $G_\mathcal{A}$.

2. For each strongly connected component $V \subseteq Q$, check whether

   2.1. $N^+_{G_\mathcal{A}}(V) = \emptyset$ and

   2.2. $V \subseteq A$ or $V \subseteq Q \setminus A$.

For the correctness, we observe that $\mathcal{A}$ has a sink if and only if there exists a strongly connected component $V \subseteq Q$ with $N^+_{G_\mathcal{A}}(V) = \emptyset$. Then the asymptotic density of L is determined by checking $V \subseteq A$ or $V \subseteq Q \setminus A$ by Theorem 3.7. For the running time, the first step can be implemented to run in time $O(n + n|\Sigma|) = O(n)$ by [Tar72]. Afterwards, for each of the at most n strongly connected components only constant-time accesses to the adjacency of $G_\mathcal{A}$ and accepting states of $\mathcal{A}$ are necessary. □

Unranked Trees. There is a well-known correspondence between binary trees in our sense (with left and right children) and forests of unranked trees, see, for example, Section 2.3.2 of [Knu97]. Every unranked tree T can be uniquely encoded by a binary tree $T^\flat$, and the binary trees $T^\flat$ obtained in this way are exactly those in which the root does not have a right child (i.e. $T^\flat_r = \emptyset$). Again there are various approaches to defining regular languages of unranked trees (such as definability in monadic second-order logic), all of which are equivalent to saying that a set L of unranked trees is regular if the language

$$L^\flat \coloneqq \{T^\flat \mid T \in L\}$$

of binary trees is regular in our sense. Theorem 3.7 therefore gives an exact and decidable characterisation of regular languages of unranked trees. However, this gives a necessary and sufficient condition on $L^\flat$ for when a regular language L is sparse. We now show that in fact:

Theorem 3.10. *A regular language L of unranked trees is sparse if, and only if, some unranked tree S does not appear as a subtree of any tree $T \in L$.*

Proof. Let S be an unranked tree with root label $a \coloneqq \lambda_S(\epsilon) \in \Sigma$. Then S is a subtree of an unranked tree T if, and only if, $T^\flat$ has a node labelled a whose left subtree is exactly $S^\flat_l$, the left subtree of the root of $S^\flat$. (Note that $S^\flat$, being the encoding of an unranked tree, has $S^\flat_r = \emptyset$.) Let us say that $S^\flat_l$ is an a-left subtree of $T^\flat$ in this situation.

Now if $\mathrm{conc}_a(S^\flat_l, \emptyset) \preceq T$ then in particular $S^\flat_l$ is an a-left subtree of T, and with Theorem 3.1 we get that

$$\mathbb{P}_{\lim}(\{T^\flat \in \mathcal{B} \mid S^\flat_l \text{ is an } a\text{-left subtreee of } T^\flat\}) = 1$$

for every $S^\flat_l \in \mathcal{B}$. Similarly, it is easy to adapt the statements and proofs of Lemma 3.4 and Theorem 3.5 to the case that $S^\flat_l$ is an a-left subtree of $T^\flat$. □

Proving Non-Regularity. In another direction, Theorem 3.7 shows a sufficient condition for the non-regularity of tree languages:

Corollary 3.11. *Let $L \subseteq \mathcal{B}^\Sigma$ be a tree language with $\mathbb{P}_{\lim}(L) = 0$. If L does not have a forbidden subtree, then L is not regular.*

Example 3.12. We call a binary tree $T \in \mathcal{B}^\Sigma$ *symmetric* if the left and right subtree obtained from the root are isomorphic. Let L_{sym} be the set of all symmetric binary trees, then for every $a \in \Sigma$ and $S \in \mathcal{B}^\Sigma$ we have $\mathrm{conc}_a(S, S) \in L_{\mathrm{sym}}$ and thus L_{sym} does not have a forbidden subtree. However, for the asymptotic density of L_{sym} we get $|L_{\mathrm{sym}} \cap \mathcal{B}^\Sigma_{2n}| = 0$ and

$$\lim_{n\to\infty} \frac{|L_{\mathrm{sym}} \cap \mathcal{B}^\Sigma_{2n+1}|}{|\mathcal{B}^\Sigma_{2n+1}|} = \lim_{n\to\infty} \frac{C_n |\Sigma|^{n+1}}{C_{2n+1}|\Sigma|^{2n+1}} = \lim_{n\to\infty} \frac{C_n}{C_{2n+1}|\Sigma|^n} = 0,$$

which yield $\mathbb{P}_{\lim}(L_{\mathrm{sym}}) = 0$. By Corollary 3.11 we infer that L_{sym} is not a regular tree language.

Random Binary Search Trees. Another prominent model for random binary trees are *random binary search trees*. These trees appear in the analysis of algorithms such as Quicksort and Find, cf. [Dev86]. A random binary search tree on n nodes is obtained by taking a root and appending to it a left subtree of size k and a right subtree of size $n-1-k$ independently, where k is chosen uniformly at random from $\{0, \ldots, n-1\}$. Formally, we let $\mathbb{P}^{\mathrm{bst}}_1 : \mathcal{B}^\Sigma_1 \to [0,1]$, $T \mapsto \frac{1}{|\Sigma|}$ and for $n > 1$ and $T \in \mathcal{B}^\Sigma_n$ set $\mathbb{P}^{\mathrm{bst}}_n(T) = \frac{1}{n|\Sigma|}\mathbb{P}^{\mathrm{bst}}_{|T_l|}(T_l)\mathbb{P}^{\mathrm{bst}}_{|T_r|}(T_r)$. For the asymptotic probability of a language $L \subseteq \mathcal{B}^\Sigma$ we again set $\mathbb{P}^{\mathrm{bst}}_{\lim}(L) \coloneqq \lim_{n\to\infty} \mathbb{P}^{\mathrm{bst}}_n(L)$, given the limit exists.

The characterisation from Theorem 3.7 however does not immediately hold for non-uniform probability measures. In the following, we consider random binary search trees and show that there are tree languages with asymptotic probability 0 which do not admit our characterisation.

Lemma 3.13. *Let $L \subseteq \mathcal{B}^\Sigma$ and $T \in \mathcal{B}^\Sigma$. Then $\mathbb{P}^{bst}_{\lim}(LT) = 0$.*

Proof. We use the independence condition in the definition of the binary search tree distribution to obtain the following:

$$\begin{aligned}
\mathbb{P}^{\mathrm{bst}}_n(LT \cap \mathcal{B}^\Sigma_n) &= \sum_{S \in LT \cap \mathcal{B}^\Sigma_n} \mathbb{P}^{\mathrm{bst}}_n(S) \\
&= \sum_{S \in L \cap \mathcal{B}^\Sigma_{n-1-|T|}} \sum_{a \in \Sigma} \mathbb{P}^{\mathrm{bst}}_n(\mathrm{conc}_a(S,T)) \\
&= \sum_{S \in L \cap \mathcal{B}^\Sigma_{n-1-|T|}} \sum_{a \in \Sigma} \frac{1}{n|\Sigma|} \mathbb{P}^{\mathrm{bst}}_{|T|}(T) \mathbb{P}^{\mathrm{bst}}_{|S|}(S) \\
&= \sum_{S \in L \cap \mathcal{B}^\Sigma_{n-1-|T|}} \frac{1}{n} \mathbb{P}^{\mathrm{bst}}_{|T|}(T) \mathbb{P}^{\mathrm{bst}}_{|S|}(S) \\
&= \frac{1}{n} \mathbb{P}^{\mathrm{bst}}_{|T|}(T) \sum_{S \in L \cap \mathcal{B}^\Sigma_{n-1-|T|}} \mathbb{P}^{\mathrm{bst}}_{|S|}(S) \\
&= \frac{1}{n} \mathbb{P}^{\mathrm{bst}}_{|T|}(T) \mathbb{P}^{\mathrm{bst}}_{n-1-|T|}(L \cap \mathcal{B}^\Sigma_{n-1-|T|}) \leq \frac{1}{n}.
\end{aligned}$$

Taking the limit yields the desired result. □

Example 3.14. Consider the tree language R that consists of all Σ-labelled binary trees T with $T_l = \{\varepsilon\}$, i.e., empty left subtree. The language R is regular because the automaton $\mathcal{A} = (\{q_0, q_1, q_2\}, \{a\}, \Delta, \{q_2\})$ with

$$\Delta := \{(\bot, \bot, a, q_0)\} \cup \{(l, r, a, q_1) \mid r \neq q_0, (l, r) \neq (\bot, \bot)\} \cup \{(l, q_0, a, q_2) \mid l \in Q \cup \{\bot\}\}$$

recognises R.

By Lemma 3.13 we have $\mathbb{P}^{\mathrm{bst}}_{\lim}(R) = 0$, but on the other hand, for all $S \in \mathcal{B}^\Sigma$ it holds that $\{T \in \mathcal{B}^\Sigma \mid S \preceq T\} \not\subseteq \mathcal{B}^\Sigma \setminus L$ since every tree might occur as a subtree in a right subtree from R. Also, the automaton $\mathcal{A}$ does not have a sink.

4 Infinite Words

We first review Sin'ya's result and Koga's simplified proof [Kog19] of it and then see how it can be extended to infinite words.

Definition 4.1. For a language $L \subseteq \Sigma^*$, the *infix language* $\mathrm{infix}(L)$ is defined as

$$\mathrm{infix}(L) := \{w \in \Sigma^* \mid xwy \in L \text{ for some } x, y \in \Sigma^*\}. \tag{2}$$

The language is said to be *infix complete* if $\mathrm{infix}(L) = \Sigma^*$.

In [Sin15], Sin'ya proved that a regular language L has asymptotic density strictly larger than 0 if and only if it is infix complete. Koga's simplified proof in [Kog19] hinges on the fact that for regular languages, the length of the prefix x and the suffix y in (2) may be bounded uniformly in L, independent of w. We need a slight strengthening of this in that the prefix x may actually be assumed to depend only on L, not on v. We prove this by giving an equivalent condition on DFAs accepting the language L:

Lemma 4.2. *Let $\mathcal{A} = (\Sigma, Q, q_0, \delta, A)$ be a deterministic finite automaton in which every state is reachable. Then $L(\mathcal{A})$ is infix complete if, and only if, some closed reachability class contains an accepting state. In this case there is a word $x \in \Sigma^*$ and a $k \geq 0$ such that for every $v \in \Sigma^*$ there is a $y \in \Sigma^{\leq k}$ with $xvy \in L$.*

Proof. Let us first assume that some closed reachability class C contains an accepting state $q \in C \cap A$. Since all states are assumed to be reachable, there is a string $x \in \Sigma^*$ with $\hat{\delta}(q_0, x) = q$. Now for every $v \in \Sigma^*$, $q_v \coloneqq \hat{\delta}(q, v) \in C$, and because q is reachable from every state in C there is a $y_v \in \Sigma^*$ with $\hat{\delta}(q_v, y_v) = q$ and therefore

$$\hat{\delta}(q_0, xvy_v) = \hat{\delta}(q, vy_v) = \hat{\delta}(q_v, y_v) = q \in A,$$

so $xvy \in L(\mathcal{A})$. Since C is finite and the string y depends only on $q_v \in C$ we may assume $|y_v| \leq k$ for some k depending only on $\mathcal{A}$.

For the converse direction, we adversarially construct a string v such that for every $q \in Q$, $\hat{\delta}(q, v)$ is an element of some closed reachability class. Then for all $x, y \in \Sigma^*$, $\hat{\delta}(q, xvy)$ is an element of a closed reachability class, and if no such class contains an accepting state we get $xvy \notin L(\mathcal{A})$, so $L(\mathcal{A})$ is not infix complete.

To construct v we first pick, for every state $q \in Q$, a string $v_q \in \Sigma^*$ such that $\hat{\delta}(q, v_q)$ is an element of some closed reachability class (such a v_q must exist because Q is finite). We enumerate the states as $Q = \{q_0, \ldots, q_r\}$ and set

$$v_0 \coloneqq \varepsilon \quad \text{and} \quad v_{i+1} \coloneqq v_i v_{\hat{\delta}(q_i, v_i)} \text{ for } i = 0, \ldots, r.$$

Then v_{r+1} has the desired property: If the automaton starts in some state q_i, then v_i takes it to some state $q = \hat{\delta}(q_i, v_i)$, and from there $v_{\hat{\delta}(q_i, v_i)} = v_q$ takes it to some closed reachability class C. Since this can not be left, also $\hat{\delta}(q_i, v_{r+1}) \in C$. □ □

Corollary 4.3. *Given a DFA $\mathcal{A}$, it is decidable whether $L(\mathcal{A})$ is infix complete.*

In fact, for a regular language L the language $\mathrm{infix}(L)$ is again regular, and one can easily compute a DFA for $\mathrm{infix}(L)$ given a DFA for L.

An ω-word $\xi \in \Sigma^\omega$ is called *rich* if for every $v \in \Sigma^*$ there are $x \in \Sigma^*$ and $y \in \Sigma^\omega$ such that $\xi = xvy$ [Sta98]. Our main result on the density of regular ω-languages now reads:

Theorem 4.4. *Let $L \subseteq \Sigma^\omega$ be a regular ω-language. Then $\mathbb{P}(L) > 0$ if and only if L contains a rich ω-word. Moreover, given a suitable description of L it is decidable whether $\mathbb{P}(L) > 0$ or not.*

Proof. Let $L = \bigcup_i U_i V_i^\omega$ with regular $U_i, V_i \subseteq \Sigma^*$. Obviously L contains a rich ω-word if, and only if, one of the languages $U_i V_i^\omega$ does, which is the case if, and only if, V_i^ω contains such a word. On the other hand, $\mathbb{P}(L) \leq \sum_{i=1}^k \mathbb{P}(U_i V_i^\omega)$, so $\mathbb{P}(L) > 0$ if and only if $\mathbb{P}(U_i V_i^\omega) > 0$ for some i. Therefore without loss of generality we may assume that $L = UV^\omega$ for regular $U, V \subseteq \Sigma^*$.

V^ω contains a rich ω-word if and only if V^* is infix complete: Since Σ^* is countable, if V^* is infix complete we may take a word $w_v = xvy \in V$ for each $v \in \Sigma^*$ and concatenate these words to get a rich ω-word in V^ω. On the other hand, if $\xi \in V^\omega$ is rich, then every $w \in \Sigma^*$ is contained in some finite prefix of ξ, which in turn is a prefix of some word in V^*.

We now show the following: If $U, V \subseteq \Sigma^*$ are regular languages and V^* is infix complete there is a word $x \in \Sigma^*$ such that

$$\mathbb{P}\big(\zeta \in L \mid x \preceq \zeta\big) = 1 \qquad \text{or, equivalently,} \qquad \mathbb{P}\big(\{\zeta \in \Sigma^\omega \mid x \preceq \zeta\} \setminus L\big) = 0$$

Since $(V^*)^\omega = V^\omega$ we may actually assume that V itself is infix complete. Pick a DFA $\mathcal{A} = (Q, q_0, \delta, A)$ with $L(\mathcal{A}) = V$. By Lemma 4.2 there is a closed reachability class $C \subseteq Q$ that contains an accepting state $\tilde{q} \in C \cap A$. Let $w = a_1 \cdots a_\ell \in \Sigma^*$ be any word such that $\hat{\delta}(q_0, w) \in C$. We define a new automaton $\mathcal{A}' = (\Sigma, Q', q_0, \delta', A)$ by $Q' \coloneqq Q \cup \{q'_1, \ldots, q'_\ell\}$ and

$$\begin{aligned}
\delta'(q, a) &= \delta(q, a) \text{ if } q \notin \{\tilde{q}, q'_1, \ldots, q'_\ell\},\\
\delta'(\tilde{q}, a) &= \begin{cases} q_1 & \text{if } a = a_1 \\ \delta(\tilde{q}, a) & \text{if } a \neq a_1 \end{cases}\\
\delta'(q'_i, a) &= \begin{cases} q'_{i+1} & \text{if } a = a_{i+1} \text{ and } i < \ell \\ \hat{\delta}(\tilde{q}, a_1 \cdots a_i a) & \text{if } a \neq a_{i+1} \end{cases}\\
\delta'(q'_\ell, a) &= \delta(\hat{\delta}(q_0, w), a)
\end{aligned}$$

Then $C \cup \{q'_1, \dots, q'_\ell\}$ is a closed reachability class in $\mathcal{A}'$ and if $\zeta = z_0 z_1 \cdots \in \Sigma^\omega$ is an ω-word such that $w \preceq \zeta$ and $\hat{\delta}'(q_0, z_0 \cdots z_k) = q'_\ell$ for infinitely many $k \geq 0$ (i.e. when reading the ω-word ζ, the automaton $\mathcal{A}'$ passes through the state q'_ℓ infinitely often), then $\zeta \in V^\omega$. In fact, let $k_1 < k_2 < \cdots$ be chosen such that $\hat{\delta}'(q_0, z_0 \cdots z_{k_i}) = q'_\ell$ for every $i \geq 1$. Then each of the subwords

$$z_0 \cdots z_{k_1 - \ell}, \quad z_{k_1 - \ell + 1} \cdots z_{k_2 - \ell}, \quad z_{k_2 - \ell + 1} \cdots z_{k_3 - \ell}, \quad \cdots$$

is in V, because it starts with the prefix w and then takes the automaton $\mathcal{A}$ to the (accepting) state $\tilde{q}$.

Now, if $\zeta = \zeta_1 \zeta_2 \zeta_3 \cdots$ is a random ω-word, the random variables $(X_t)_{t \geq 0}$ with $X_t = \hat{\delta}'(q_0, \zeta_1 \cdots \zeta_t)$ are a Markov chain with state space Q', started in q_0 and with transition probabilities $p_{q,q'} = |\Sigma|^{-1}$ if $\delta(q, a) = q'$ for some $a \in \Sigma$, and $p_{q,q'} = 0$ otherwise. In particular, the reachability relation $\to$ on the state space of this Markov chain is the same as the reachability relation $\rightsquigarrow$ of the automaton $\mathcal{A}'$, and the closed classes of the Markov chain are exactly the closed reachability classes of $\mathcal{A}'$. Thus q'_ℓ is recurrent by Thm. 2.3, and with Cor. 2.4 we get

$$\mathbb{P}(X_t = q'_\ell \text{ infinitely often} \mid w \preceq \zeta) = \mathbb{P}(X_t \in C \text{ for some } t \geq 0 \mid w \preceq \zeta) = 1,$$

because if $w \preceq \zeta$ then $X_\ell = \hat{\delta}(q_0, w) = \tilde{q} \in C$. Finally, if $u \in U$ is any word in U then $\mathbb{P}\big(\zeta \in L \mid uw \preceq \zeta\big) = 1$. □

5 Conclusion

We gave decidable characterisations for sparseness of regular tree languages and of regular ω-languages, both in terms of excluded subtrees and in terms of automata accepting these languages.

By [NPS23], sparseness is decidable also for regular languages of *infinite* trees, allowing for probabilistic model checking not just for linear temporal logic, but also for computation tree logic (CTL), opening the route to many further applications in model checking (cf. [BKL08, Ch. 6]). However, Niwinski et al. solve the more general problem of exactly computing the measure of a regular infinite tree language, at prohibitively high computational cost. Focussing on sparseness is likely to allow for more efficient algorithms, as exemplified by the linear time algorithm from Corollary 3.9.

Several known graph properties, such as being series-parallel or, more generally, having bounded tree-width, imply that graphs with these properties can be encoded in labelled trees in such a way that the original graph can be MSO interpreted in the tree (cf. [FG06, Ch. 11.4]). Our results on sparseness of regular tree languages can therefore be translated into sparseness conditions for MSO-definable properties of graphs in these graph classes. However, since a given graph may in general be interpretable in many different trees, this yields sparseness with respect to some non-uniform notion of density, prompting for a closer investigation.

Another interesting direction for future work would be more general probabilistic models. While our methods (as well as those in [CY95] and [NPS23]) easily generalise from independent letters to Markov chains, it is not immediately clear if this can be done also for strings or trees generated by hidden Markov models (HMMs, cf. [BJ09]) because in this case, reachability classes of the automaton are no longer the same as communicating classes of the resulting Markov chain.

References

[AKNS18] Albert Atserias, Stephan Kreutzer, and Marcos Noy Serrano. On zero-one and convergence laws for graphs embeddable on a fixed surface. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018): July 9-13, 2018, Prague, Czech Republic*, pages 1–14. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ICALP.2018.116.

[AKST19] Kazuyuki Asada, Naoki Kobayashi, Ryoma Sin'ya, and Takeshi Tsukada. Almost every simply typed lambda-term has a long beta-reduction sequence. *Log. Methods Comput. Sci.*, 15(1), 2019. doi:10.23638/LMCS-15(1:16)2019.

[BJ09] Yoon Byung-Jun. Hidden Markov models and their applications in biological sequence analysis. *Current Genomics*, 10(6):402–415, 2009. doi:10.2174/138920209789177575.

[BKL08] C. Baier, J.P. Katoen, and K.G. Larsen. *Principles of Model Checking*. MIT Press, 2008.

[BS09] Michael Benedikt and Luc Segoufin. Towards a characterization of order-invariant queries over tame graphs. *J. Symb. Log.*, 74(1):168–186, 2009. doi:10.2178/jsl/1231082307.

[Bü60] J. Richard Büchi. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6(1-6):66–92, 1960. doi:https://doi.org/10.1002/malq.19600060105.

[Bü90] J. Richard Büchi. *On a Decision Method in Restricted Second Order Arithmetic*, pages 425–435. Springer New York, New York, NY, 1990. doi:10.1007/978-1-4613-8928-6_23.

[CY95] Costas Courcoubetis and Mihalis Yannakakis. The complexity of probabilistic verification. *Journal of the ACM (JACM)*, 42(4):857–907, 1995. doi:10.1145/210332.210339.

[Dev86] Luc Devroye. A note on the height of binary search trees. *Journal of the ACM (JACM)*, 33(3):489–498, 1986. doi:10.1145/5925.5930.

[DK19] Anuj Dawar and Eryk Kopczynski. Logical properties of random graphs from small addable classes. *Log. Methods Comput. Sci.*, 15(3), 2019. doi:10.23638/LMCS-15(3:4)2019.

[Don70] John Doner. Tree acceptors and some of their applications. *J. Comput. Syst. Sci.*, 4(5):406–451, 1970. doi:10.1016/S0022-0000(70)80041-1.

[EF99] Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite Model Theory*. Perspectives in Mathematical Logic. Springer, 2nd edition, 1999. doi:10.1007/3-540-28788-4.

[Elg61] Calvin C. Elgot. Decision problems of finite automata design and related arithmetics. *Transactions of the American Mathematical Society*, 98(1):21–51, 1961. doi:10.2307/1993511.

[Fag76] Ronald Fagin. Probabilities on finite models. *The Journal of Symbolic Logic*, 41(1):50–58, 1976. doi:10.2307/2272945.

[FG06] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006. doi:10.1007/3-540-29953-X.

[FS09] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009. doi:https://doi.org/10.1017/CBO9780511801655.

[GKLT69] Y.V. Glebskij, D.I. Kogan, M.I. Liogon'kij, and V.A. Talanov. Range and degree of realizability of formulas in the restricted predicate calculus. *Cybernetics*, 5:142–154, 1969. doi:10.1007/BF01071084.

[Knu97] Donald E. Knuth. *The Art of Computer Programming: Fundamental Algorithms*, volume I. Addison-Wesley, 3rd edition, 1997.

[Kog19] Toshihiro Koga. On the density of regular languages. *Fundam. Informaticae*, 168(1):45–49, 2019. doi:10.3233/FI-2019-1823.

[McC02] Gregory L. McColm. Mso zero-one laws on random labelled acyclic graphs. *Discrete Mathematics*, 254(1):331–347, 2002. doi:https://doi.org/10.1016/S0012-365X(01)00375-2.

[MZ21] Y.A. Malyshkin and M.E. Zhukovskii. Mso 0-1 law for recursive random trees. *Statistics & Probability Letters*, 173:109061, 2021. doi:https://doi.org/10.1016/j.spl.2021.109061.

[Nor97] James R. Norris. *Markov Chains*. Cambridge University Press, 1997. doi:10.1017/CBO9780511810633.

[NPS23] Damian Niwiński, Paweł Parys, and Michał Skrzypczak. The probabilistic Rabin tree theorem*. In *2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13, 2023. doi:10.1109/LICS56636.2023.10175800.

[Sin15] Ryoma Sin'ya. An automata theoretic approach to the zero-one law for regular languages: Algorithmic and logical aspects. In Javier Esparza and Enrico Tronci, editors, *Proceedings of the Sixth International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2015, Genoa, Italy, 21-22nd September 2015*, volume 193 of *EPTCS*, pages 172–185, 2015. doi:10.4204/EPTCS.193.13.

[Spe13] Joel Spencer. *The strange logic of random graphs*, volume 22. Springer Science & Business Media, 2013. doi:10.1007/978-3-662-04538-1.

[Sta97] Ludwig Staiger. ω-languages. In *Handbook of Formal Languages: Volume 3 Beyond Words*, pages 339–388. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997. doi:10.1007/978-3-642-59126-6_6.

[Sta98] Ludwig Staiger. Rich ω-words and monadic second-order arithmetic. In Mogens Nielsen and Wolfgang Thomas, editors, *Computer Science Logic*, pages 478–490, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. doi:10.1007/BFb0028032.

[Tar72] Robert Endre Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Comput.*, 1(2):146–160, 1972. doi:10.1137/0201010.

[Tho97] Wolfgang Thomas. Languages, automata, and logic. In *Handbook of Formal Languages: Volume 3 Beyond Words*, pages 389–455. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997. doi:10.1007/978-3-642-59126-6_7.

[Tra61] Boris A. Trakhtenbrot. Finite automata and the logic of single-place predicates. *Dokl. Akad. Nauk SSSR*, 140(2):326–329, 1961. URL: https://www.mathnet.ru/eng/dan25511.

[TW68] James W. Thatcher and Jesse B. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Math. Syst. Theory*, 2(1):57–81, 1968. doi:10.1007/BF01691346.

[Wil91] David Williams. *Probability with Martingales*. Cambridge University Press, 1991. doi:10.1017/CBO9780511813658.