

Absorbing Patterns in *BST*-Like Expression-Trees

Florent Koechlin ✉

Laboratoire d'Informatique Gaspard-Monge, Université Gustave Eiffel, Marne-la-Vallée, France

Pablo Rotondo ✉

Laboratoire d'Informatique Gaspard-Monge, Université Gustave Eiffel, Marne-la-Vallée, France

Abstract

In this article we study the effect of simple semantic reductions on random *BST*-like expression-trees. Such random unary-binary expression-trees are often used in benchmarks for model-checking tools. We consider the reduction induced by an absorbing pattern for some given operator \otimes , which we apply bottom-up, producing an equivalent (and smaller) tree-expression. Our main result concerns the expected size of a random tree, of given input size $n \rightarrow \infty$, after reduction. We show that there are two different thresholds, leading to a total of five regimes, ranging from no significant reduction at all, to almost complete reduction. These regimes are completely characterized according to the probability of the absorbing operator. Our results prove that random *BST*-like trees have to be considered with care, and that they offer a richer range of behaviours than uniform random trees.

2012 ACM Subject Classification Mathematics of computing \rightarrow Discrete mathematics

Keywords and phrases *BST* trees, absorbing pattern, reduction, analytic combinatorics

Digital Object Identifier 10.4230/LIPIcs.STACS.2021.48

Funding *Pablo Rotondo*: partially supported by the Projet RIN Alenor (Regional Project from French Normandy).

Acknowledgements The authors would like to thank Arnaud Carayol and Cyril Nicaud for their very useful advice and remarks in the writing of this paper.

1 Introduction

There are two main ways to evaluate the performances of an algorithm or of its implementation: a theoretical one studying its complexity, and a more practical one, using benchmarks. On the theoretical side, it is classical to study the worst-case performances, which only gives a partial view of the practical usability of an algorithm. For instance, bubblesort and quicksort have the same worst-case complexity but perform very differently in practice to the extent that quicksort is actually implemented in some standard libraries. Average complexity tries to remedy this problem by considering a more appropriate probabilistic model on the inputs. The problem with this approach is that one must find a distribution simple enough to be mathematically tractable whilst being complex enough to model accurately the *real life distribution*. In the literature, this latter aspect is often relegated to a second place by theorists, as even simple algorithms may be hard to analyze for the uniform distribution.

Practical approaches consist in executing the tools and measuring directly their performances. These benchmarks are performed on real-world test cases, when possible. These are often complemented with randomly generated ones, which is an easy way of generating test cases of arbitrary sizes. Note that in this setting one is free to choose more mathematically complex distributions for the inputs, provided that they are fast to generate, and reasonably close to real-life examples.

In this paper we concentrate on algorithms manipulating tree-like expressions. Tree-like expressions are ubiquitous in computer science, for describing regular languages, boolean formulas, LTL formulas, \dots In Figure 1 we give several examples of such expression trees.



© Florent Koechlin and Pablo Rotondo;

licensed under Creative Commons License CC-BY 4.0

38th International Symposium on Theoretical Aspects of Computer Science (STACS 2021).

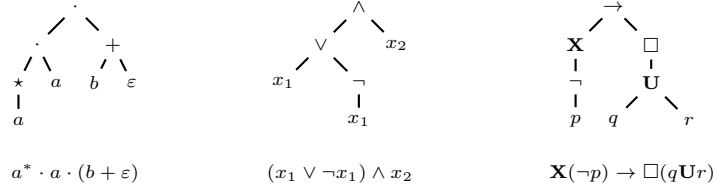
Editors: Markus Bläser and Benjamin Monmege; Article No. 48; pp. 48:1–48:15

Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Note that this representation is purely syntactical and might be redundant; several trees can have the same semantics, i.e., represent the same object. For example, the logical formula in Figure 1 is equivalent to x_2 .



■ **Figure 1** Three examples of expression trees. From left to right: a regular expression, a logical formula and an LTL formula.

In the absence of information on the real-life distribution for the algorithms, the uniform distribution is commonly considered: it appears a natural choice as in some sense it maximizes the coverage of possible inputs. Recently it was shown in [11, 12] that the uniform distribution is not relevant in benchmarks for tools manipulating tree-expressions. The authors proved that even if this distribution offers a good coverage of the tree-expressions, their coverage is poor when it comes to the objects represented by these tree-expressions, in the presence of simple simplifications. More precisely, the authors consider tree expressions with an operator \otimes which admits a particular fixed tree \mathcal{P} , called the *absorbing pattern*, such that $\mathcal{P} \otimes t$, $t \otimes \mathcal{P}$ and \mathcal{P} represent the same object. This situation occurs in most natural examples. For instance, **False** is absorbing for \wedge , $(a + b)^*$ is absorbing for $+$ in regular expressions, 0 is absorbing for \times in arithmetic expressions, etc. In the presence of an absorbing pattern, one can reduce a tree-expression in a bottom-fashion in order to remove all occurrences of a tree-expression of the form $\mathcal{P} \otimes t$ or $t \otimes \mathcal{P}$. This reduction can be performed in linear time and preserves the object represented by the tree-expression. Surprisingly, the authors proved that if we draw uniformly at random a tree-expression of size n the expected size of its reduced tree-expression is bounded by a constant. Hence using the uniform distribution in a benchmark might only be testing the ability of the tool to perform simple reductions on expressions.

Looking at the random generators used in most benchmarks for model-checking tools dealing with tree-expressions, we see that the distribution produced is not the uniform one. In the case of random LTL formulas, Algorithm 1 presents the algorithm used by the tool **LTL-to-Büchi translator testbench (lbt)** of TCS [17] (see also [4] and **Spot** [6] for other examples). They are all based on well-known distributions in combinatorics called *BST-like* distributions [7] as they are strongly related to random binary search trees.

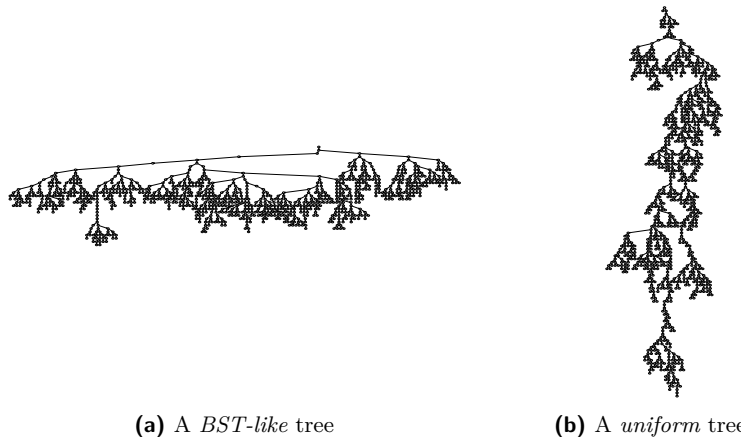
The procedure used to generate such random expression trees of size n , where n is the number of nodes, is the following one:

- (0) assign a probability distribution for the operators, and another one for the leaves;
- (1) if $n = 1$ then draw a random leaf;
- (2) if $n = 2$, then draw an operator of arity 1 following their probabilities, and a random leaf;
- (3) if n is greater than 2, draw a random operator for the root. If the operator is unary, proceed to build a subtree of size $n - 1$. If the operator is binary, draw first uniformly the sizes for the left and right subtrees (so that they add to $n - 1$), and recursively generate these subtrees.

■ **Algorithm 1** The pseudo-code used in `lbt` [17, p.46] to draw a random LTL formula.

```

function RandomFormula( $n$ ):
  if  $n = 1$  then
     $p :=$  random symbol in  $AP \cup \{\top, \perp\}$ ;
    return  $p$ ;
  else if  $n = 2$  then
     $op :=$  random operator in  $\{\neg, \mathbf{X}, \square, \diamond\}$ ;
     $f :=$  RandomFormula(1);
    return  $op f$ ;
  else
     $op :=$  random operator in  $\{\neg, \mathbf{X}, \square, \diamond, \wedge, \vee, \rightarrow, \leftrightarrow, \mathbf{U}, \mathbf{R}\}$ ;
    if  $op$  in  $\{\neg, \mathbf{X}, \square, \diamond\}$  then
       $f :=$  RandomFormula( $n - 1$ );
      return  $op f$ ;
    else
       $x :=$  uniform integer in  $[1, n - 2]$ ;
       $f_1 :=$  RandomFormula( $x$ );
       $f_2 :=$  RandomFormula( $n - x - 1$ );
      return  $(f_1 op f_2)$ ;
```



■ **Figure 2** The expected height of a random BST tree (on the left) of size n is $\Theta(\log(n))$ whereas for a uniform tree (on the right) of size n , it is $\Theta(\sqrt{n})$.

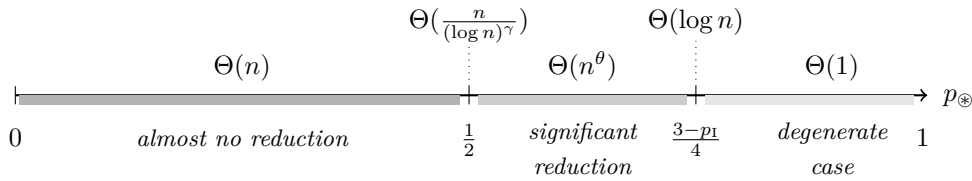
The resulting distribution over the trees with n nodes is not uniform. In fact, the shape of a typical tree drawn from the BST-like distribution differs greatly from that of a tree drawn from the uniform distribution [5, 8]. It can be seen by comparing Figure 2a and Figure 2b. This difference is also apparent when it comes to the average behaviour of algorithms. It was shown in [14] that the Glushkov automaton (a.k.a., the position automaton) of a regular expression under a BST-like distribution has an average of $\Theta(n^2)$ transitions, in stark contrast to the case of the uniform distribution, where it was previously showed [13] that the average is $\Theta(n)$. Observe that for the uniform distribution, if we reduce the expression (according to the absorbing pattern $(a + b)^*$ for $+$) first, the expected size of the Glushkov automaton is in fact bounded by a constant, as a consequence of [11, 12].

It seems likely that the choice of the *BST*-like distributions in benchmark is motivated by its greater flexibility to model real-world distributions (i.e. by playing with the operator probabilities) and also because of its very efficient generation procedure.

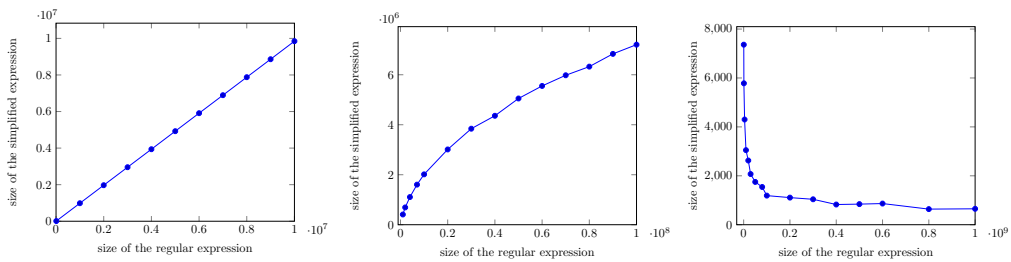
Seeing the flaw of the uniform distribution for tree-expressions discovered in [11, 12], it is natural to wonder if the *BST*-like distributions suffer from the same short-comings. This question is the starting point of the work present in this paper. We assume the existence of an absorbing pattern \mathcal{P} for some operator \otimes and study the expected size of the tree-expression after reduction¹.

Our main result paints a complete picture of the possible asymptotic behaviour of the expected size after reduction as the original size n tends to infinity. We show that there are two different thresholds, leading to a total of five regimes, depending on the probability p_{\otimes} of the absorbing operator and the probability of drawing a unary operator p_1 . The main regimes are shown experimentally in Figure 3.

► **Theorem.** Consider a family of expression trees defined from unary and binary operators. Suppose there is a tree pattern \mathcal{P} , of size at least² 3, that is absorbing for a distinguished operator \otimes . We consider the simplification algorithm that consists in inductively changing a \otimes -node by \mathcal{P} whenever one of its children can be simplified into \mathcal{P} . Then the expected size of a random *BST*-like tree after simplification has an asymptotic behaviour given by the following cases, depending on the probability p_{\otimes} of the absorbing operator:



There are two critical points $p_{\otimes} = 1/2$ and $p_{\otimes} = (3 - p_1)/4$, the latter depending on the probability p_1 of drawing a unary operator. This gives a total of five regimes spanning the spectrum from almost no reduction $\Theta(n)$ to complete reduction $\Theta(1)$. The exponents γ and θ are given by $\gamma = \frac{2}{1-p_1}$ and $\theta = 1 - \frac{4p_{\otimes}-2}{1-p_1}$ respectively.



■ **Figure 3** The three main regimes observed experimentally on regular expressions on two letters, with 10 000 samples for each size: (from left to right) linear ($p_+ = p_* = p = 1/3$), sublinear ($p_+ = 19/29, p_* = p = 5/29$) and constant ($p_+ = 8/10, p_* = p = 1/10$).

¹ Particular cases where these and many more reductions are available have been studied in the literature [2, 3, 10]. However, these consider a very particular example, namely the \wedge/\vee -trees with $p_{\wedge} = p_{\vee} = \frac{1}{2}$.

² This restriction is not a real constraint. For $|\mathcal{P}| \leq 2$ it is easy to build from \mathcal{P} a larger absorbing pattern and our result then applies. See the discussion at the end of Section 2.2.

Methods. To obtain our results we employ techniques from the framework of Analytic Combinatorics [9]. The recursive procedure used to produce a random BST-like tree naturally leads in turn to a recurrence for the probabilities of interest. We encode the probabilities into ordinary generating functions, and the recurrences lead formally to differential equations (of Riccati type) for the expected value of the size after reduction. This first part is purely symbolic. At this point, as is usual in Analytic Combinatorics, we see our generating functions as power series on the complex plane to take advantage of the powerful theory of holomorphic functions. The singularities, i.e., the points where these functions cease to be smooth, are related to the asymptotics of the coefficients. In particular, those that are closer to the origin (i.e., dominant singularities) give the leading terms. This link is formally given by the Transfer Theorem [9, Ch VI.3], which translates asymptotic equivalents for the generating functions near the singularities to asymptotics for their coefficients.

Plan of the article. In Section 2 we give precise definitions for our settings, namely random BST-like expression trees, absorbing patterns and the ensuing simplification. Section 3 gives a general overview of the techniques and gives an outline of the proof of our main Theorem. It explains in particular why we need to consider the probability of fully reducible trees first (those reducing to \mathcal{P}), in order to prove our main result. Section 4 then is devoted to these fully reducible trees, and as a side product we prove (see Thm 10) that the probability of being fully reducible tends to 0 for $p_{\otimes} \leq 1/2$ and to a positive constant otherwise. This first threshold is intimately linked with that of the main result. Finally, Section 5 completes the sketched proof of the main theorem.

Most proofs are either sketched or omitted in this extended abstract.

2 Model, definitions and probability of complete reduction

2.1 The BST-like model

Consider three non-empty sets of labels \mathcal{A}_0 , \mathcal{A}_1 , and \mathcal{A}_2 , corresponding respectively to the sets of possible leaves, unary operators and binary operators. For example, to describe the set of regular expressions on the alphabet $\{a, b\}$: $\mathcal{A}_0 = \{\varepsilon, a, b\}$, $\mathcal{A}_1 = \{\star\}$ and $\mathcal{A}_2 = \{\cdot, +\}$.

We define the family of trees on $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ in Definition 1 below. It is important to emphasize that when we say trees we actually mean planar trees throughout the article: the order of the branches does matter, hence $\bigwedge_{T_1 T_2}^{op}$ and $\bigwedge_{T_2 T_1}^{op}$ are not the same tree.

► **Definition 1** (Expression trees). *The family $\mathcal{E}(\mathcal{A})$ of expression trees over $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ is defined inductively:*

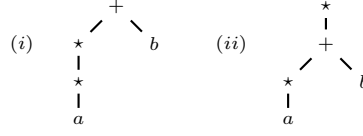
- any leaf $a_0 \in \mathcal{A}_0$ is an expression tree;
- if T is an expression tree and $op_1 \in \mathcal{A}_1$ is a unary operator, then $\bigwedge_T^{op_1} \in \mathcal{E}(\mathcal{A})$;
- if T_1, T_2 are expression trees and $op_2 \in \mathcal{A}_2$ is a binary operator, then $\bigwedge_{T_1 T_2}^{op_2} \in \mathcal{E}(\mathcal{A})$.

The size $|T|$ of an expression tree T is its number of nodes (operators and leaves).

Now that we have defined the family of expression trees, we introduce the BST-like distribution over them. For $n \in \mathbb{N}$, let \mathcal{E}_n denote the set of expression trees of size n .

First we endow the set of leaves \mathcal{A}_0 and the set of operators $\mathcal{A}_{ops} = \mathcal{A}_1 \cup \mathcal{A}_2$ with probabilities, $(p_a)_{\mathcal{A}_0}$ and $(p_{op})_{\mathcal{A}_{ops}}$. Remark then that $\sum_{a \in \mathcal{A}_0} p_a = 1$ and $\sum_{op \in \mathcal{A}_{ops}} p_{op} = 1$.

We denote by p_I the probability of picking a unary operator, i.e., $p_I = \sum_{op_1 \in \mathcal{A}_1} p_{op_1}$.



■ **Figure 4** Example of two trees of size $n = 5$, for regular expressions, having different probabilities for any choice of distribution. The probability of the trees in (i) and (ii) are $\frac{1}{3}p_+p_*p_ap_b$ and $\frac{1}{2}p_+p_*p_ap_b$ respectively.

▶ **Definition 2** (Random *BST*-like expression tree). A random *BST*-like expression tree of size $n \in \mathbb{N}_{\geq 1}$ is built recursively as follows:

- If $n = 1$, we draw a leaf from \mathcal{A}_0 according to the probability distribution $(p_a)_{a \in \mathcal{A}_0}$.
- If $n = 2$, we draw a unary operator op_1 according to the normalized probabilities $\left(\frac{1}{p_1}p_{op_1}\right)_{op_1 \in \mathcal{A}_1}$, then we draw independently a tree $a_0 \in \mathcal{A}_0$ of size 1 and return $\begin{array}{c} op_1 \\ | \\ a_0 \end{array}$.
- If $n \geq 3$, we pick an operator $\oplus \in \mathcal{A}_{ops}$ according to the distribution $(p_{op})_{op \in \mathcal{A}_{ops}}$. (★) If we obtain a unary operator $\oplus \in \mathcal{A}_1$, we produce recursively and independently a tree T of size $n - 1$ and return $\begin{array}{c} \oplus \\ | \\ T \end{array}$. (★★) If otherwise we obtain a binary operator $\oplus \in \mathcal{A}_2$, we pick the size k of the left subtree uniformly from $\{1, \dots, n - 2\}$ and produce independently two trees T_L and T_R of sizes k and $n - 1 - k$ respectively. Then we return $\begin{array}{c} \oplus \\ \wedge \\ T_L \quad T_R \end{array}$.

For Definition 2 to make sense, we assume that $p_1 > 0$. Note that otherwise (if $p_1 = 0$) we would produce no trees of size 2, or any even size. This assumption is not really a constraining one, as otherwise we would obtain similar results, but just over the odd sizes.

The procedure of Definition 2 defines a probability distribution over expression trees: the probability $\Pr_n(T)$ of a given expression tree T of size n is the probability of the algorithm in Definition 2 returning T with input n . This distribution is not uniform, as shown in Figure 4.

2.2 Absorbing pattern and reduction

We now define what we mean by an *absorbing pattern* for the family of expression trees $\mathcal{E}(\mathcal{A})$. Fix a binary operator $\otimes \in \mathcal{A}_2$ and an expression tree $\mathcal{P} \in \mathcal{E}(\mathcal{A})$. Informally, the associated simplification $\sigma = \sigma_{\mathcal{P}, \otimes}$ is defined by applying bottom up the substitution

$$\begin{array}{c} \otimes \\ \wedge \\ T_1 \quad T_2 \end{array} \rightsquigarrow \mathcal{P}, \text{ whenever } T_i = \mathcal{P} \text{ for some } i \in \{1, \dots, 2\}.$$

More precisely, we define recursively the simplification $\sigma = \sigma_{\mathcal{P}, \otimes}: \mathcal{E}(\mathcal{A}) \rightarrow \mathcal{E}(\mathcal{A})$ with absorbing pattern \mathcal{P} for the operator \otimes as follows: if $e \in \mathcal{A}_0$ we set $\sigma(e) = e$ while,

- $\sigma\left(\begin{array}{c} op_1 \\ | \\ T \end{array}\right) = \begin{array}{c} op_1 \\ | \\ \sigma(T) \end{array}$ for $op_1 \in \mathcal{A}_1$,
- $\sigma\left(\begin{array}{c} op_2 \\ \wedge \\ T_1 \quad T_2 \end{array}\right) = \begin{array}{c} op_2 \\ \wedge \\ \sigma(T_1) \quad \sigma(T_2) \end{array}$ for $op_2 \in \mathcal{A}_2$ with $op_2 \neq \otimes$,
- $\sigma\left(\begin{array}{c} \otimes \\ \wedge \\ T_1 \quad T_2 \end{array}\right) = \mathcal{P}$ if $\sigma(T_i) = \mathcal{P}$ for $i = 1$ or $i = 2$, and $\sigma\left(\begin{array}{c} \otimes \\ \wedge \\ T_1 \quad T_2 \end{array}\right) = \begin{array}{c} \otimes \\ \wedge \\ \sigma(T_1) \quad \sigma(T_2) \end{array}$ otherwise.

An expression tree $T \in \mathcal{E}(\mathcal{A})$ is said to be *fully reducible* when $\sigma(T) = \mathcal{P}$.

Henceforth we assume that our family of expression trees $\mathcal{E}(\mathcal{A})$ admits an absorbing \mathcal{P} of size $s := |\mathcal{P}|$ for a fixed binary operation $\otimes \in \mathcal{A}_2$. For technical reasons, we will suppose that $s \geq 3$. This might seem in contradiction to the fact that some leaf can be absorbing (for instance `False` with `∨`). However this is not much of a restriction since you can always

build, from an absorbing pattern \mathcal{P} of size less than 3, a new one of size more than 3 by considering $\mathcal{P}' := \bigwedge_{\mathcal{P} a}^{\otimes}$ for a leaf a . This new pattern leads to less reductions in comparison to the former one, so that our results give upperbounds for the expected size after reduction by an absorbing pattern of size less than 3, instead of exact equivalents.

3 Outline of the proof

For our proof we employ methods from the framework of Analytic Combinatorics [9]: we will represent a sequence $(a_n)_{n \geq 0}$ of coefficients by its *ordinary generating function* (OGF for short) $F(z) = \sum a_n z^n$. At first, we treat $F(z)$ as a formal object, and our goal is to obtain an equation characterizing it. Typically, in Analytic Combinatorics, this first step is done by building the studied combinatorial class from set operations, and using a toolbox to translate them into operations between generating functions. In our case it does not apply and we have to extract the equations for the generating functions from the recurrence relations satisfied by the related sequences. This approach is common when the distribution of the studied combinatorial class is not uniform (see for instance [9, 15]). Hence we begin the proof by deriving a recurrence relation satisfied by the expected size (e_n) . This relation comes from the recursive nature of the algorithm for constructing a random BST-like tree-expression.

3.1 Recurrence relations

Recurrence for the expected value

We are interested in the probabilities $p_{n,k} := \Pr_n\{T : |\sigma(T)| = k\}$ for a tree of size n to have a reduced size k . More precisely we want to obtain an equation for $e_n := \sum_k k \cdot p_{n,k}$ which is the expected size after reduction for a random tree of fixed size n , according to the BST-like distribution³. The following proposition gives the recurrence satisfied by the sequence $(e_n)_n$. It involves the probability that a tree T of size n is fully reducible:

$$\gamma_n = \Pr_n\{\sigma(T) = \mathcal{P}\}.$$

We also write $p_{\Pi} := 1 - p_{\Gamma} - p_{\otimes}$, the probability of drawing a binary operator that is not \otimes .

► **Proposition 3.** *The sequence (e_n) of expected sizes after reduction satisfies, for all $n > 1$:*

$$e_{n+1} = 1 + (s-1)\gamma_{n+1}\mathbf{1}_{n+1 \neq s} + p_{\Gamma}e_n + \frac{2p_{\Pi}}{n-1} \sum_{j=1}^{n-1} e_j + \frac{2p_{\otimes}}{n-1} \sum_{j=1}^{n-1} (e_j - s\gamma_j)(1 - \gamma_{n-j}).$$

Proof sketch. We introduce the auxiliary polynomials $F_n(u) = \sum_{k=0}^n \Pr_n\{T : |\sigma(T)| = k\}u^k$. These satisfy the recurrence

$$F_{n+1}(u) = \gamma_{n+1}\mathbf{1}_{n+1 \neq s}u^s + p_{\Gamma}uF_n(u) + u \frac{p_{\Pi}}{n-1} \sum_{j=1}^{n-1} F_j(u)F_{n-j}(u) + u \frac{p_{\otimes}}{n-1} \sum_{j=1}^{n-1} (F_j(u) - \gamma_j u^s)(F_{n-j}(u) - \gamma_{n-j} u^s).$$

³ From now on, when we write random, we implicitly mean for the BST-like distribution.

Indeed, a tree T of size $n + 1$ is either fully reducible (with probability γ_{n+1}) or not. When we pick \otimes , the new tree does not reduce to \mathcal{P} only when the subtrees are not fully reducible.

Then e_n is expressed as $e_n = F'_n(1)$. Differentiating the formula and setting $u = 1$ we obtain the recurrence for e_n , using the fact that $F_k(1) = 1$ for all k . ◀

Recurrence for the probability of full reduction

The recurrence for the expected values (e_n) in Proposition 3 depends strongly on the auxiliary sequence of probabilities $(\gamma_n)_{n \geq 1}$. Clearly, any tree starting by the absorbing operator \otimes and having a fully-reducible child is also fully reducible. Reciprocally, if a tree of size strictly bigger than s is fully reducible, then it has \otimes as a root and at least one fully reducible child. Hence the sequence (γ_n) satisfies a recurrence, which is not linear:

$$\gamma_{n+1} = p_{\otimes} \cdot \frac{1}{n-1} \sum_{k=1}^{n-1} (\gamma_k + \gamma_{n-k} - \gamma_k \gamma_{n-k}), \quad \text{for all } n \geq s. \tag{1}$$

Indeed, suppose that k is the size of the left subtree⁴, which happens with probability $\frac{1}{n-1}$. Then the probability that one of the children is fully reducible is, by inclusion-exclusion, $\gamma_k + \gamma_{n-k} - \gamma_k \gamma_{n-k}$.

In our study of the sequence $(\gamma_n)_{n \geq 1}$ we will show that it actually converges (Thm. 10). For the time being, we will just remark that if $(\gamma_n)_n$ converges, only certain values are possible for $L = \lim \gamma_n$. For this, let us recall this classical result:

► **Lemma 4 (Cèsaro-means).** *Consider a sequence $(a_n)_{n \geq 1}$ converging to a real number L . Then we have $\lim_n \frac{1}{n} \sum_{k=1}^n a_k = L$, and $\lim_n \frac{1}{n} \sum_{k=1}^n a_k a_{n+1-k} = L^2$.*

From Eq. (1) we see that $L = p_{\otimes} \cdot (2L - L^2)$. Thus the limit, if it exists, can only be 0 or $\gamma_{\infty} := 2 - 1/p_{\otimes}$. For $p_{\otimes} < 1/2$, we have $\gamma_{\infty} < 0$ and so $L = 0$. For $p_{\otimes} > 1/2$, Theorem 10 will show that $L = \gamma_{\infty}$. These limits hint at the possibility of a threshold for e_n at $p_{\otimes} = \frac{1}{2}$.

3.2 Main steps

In order to study the sequence of expected sizes $(e_n)_{n \geq 1}$ it will be necessary to study first the sequence of probabilities $(\gamma_n)_{n \geq 1}$. As announced, we introduce their generating functions:

$$A(z) := \sum_{n=0}^{\infty} \gamma_n z^n, \quad E(z) := \sum_{n=0}^{\infty} e_n z^n.$$

The proof, as is usual in Analytic Combinatorics, proceeds in two steps: a symbolic step and an analytic step. In the symbolic step we obtain appropriate equations for our generating functions, seen as purely formal power series. In our case it will be differential equations, coming from the recurrences. Then in the analytic step, the generating functions are seen as analytic functions of a complex variable. We apply the celebrated *Transfer theorem* (see [9, Ch VI.3]) to obtain the asymptotic equivalents of the sequences. The Transfer Theorem states that, under analytic conditions, an equivalent $E(z) \sim_{z \rightarrow 1} \lambda(1-z)^{-\alpha}$ with $\alpha \notin \{0, -1, -2, \dots\}$, implies $e_n \sim \lambda n^{\alpha-1} / \Gamma(\alpha)$, where Γ is Euler's Gamma-function, the generalized factorial.

⁴ We have supposed that there are trees of every possible size, which is equivalent to $p_I > 0$.

Symbolic step

The recurrence (1) for γ_n , as well as the recurrence of e_n in Proposition 3, lead naturally to ordinary differential equations for $A(z)$ and for $E(z)$. As the formal derivative of a series $F(z) = \sum a_n z^n$ is given by $F'(z) = \sum (n+1)a_n z^n$, multiplying Eq. (1) by $(n-1)z^n$ and summing will introduce derivatives. Thus we obtain a differential equation for $A(z)$, under the form of a Riccati equation, and a linear one for $E(z)$, which involves the generating function $A(z)$ as a known quantity:

$$A'(z) = (s-2)\gamma_s z^{s-1} + \left(\frac{2}{z} + 2p_{\otimes} \frac{z}{1-z}\right) A(z) - p_{\otimes} \cdot (A(z))^2, \tag{2}$$

and, for a certain function $F(x, y)$, which can be made explicit

$$E'(z) = F(z, A(z)) + \frac{1}{1-p_1 z} \left(\frac{2}{z} - p_1 + 2(1-p_1) \frac{z}{1-z} - 2p_{\otimes} A(z)\right) \cdot E(z).$$

These differential equations constitute our symbolic specifications for the generating functions $A(z)$ and $E(z)$. At this point we switch to their analytic study.

Analytic step

The equation for $E(z)$ is a first order linear ODE, as such it can be solved by the method of variation of constants⁵ [1, Th. 6.1] to obtain an explicit solution that involves $A(z)$ as a known quantity. Thus we need first to study $A(z)$ as a complex function, and in particular its domain of analyticity. Since the coefficients of $A(z)$ are probabilities $\gamma_n \in [0, 1]$, the series $A(z)$ defines an analytic function on the unit disk $|z| < 1$. However, for technical reasons we need further information regarding its domain of analyticity in order to apply the Transfer Theorem. Thus in Section 4 we are going into more detail, showing that $z = 1$ is a dominant singularity and that $A(z)$ can be extended analytically to the domain $\Omega = \mathbb{C} \setminus [1, \infty)$. We remark that then the same holds for $E(z)$.

The last hypothesis in order to apply the Transfer Theorem for $E(z)$ is its asymptotic equivalent as $z \rightarrow 1$, its dominant singularity. The solution of the ODE for $E(z)$ yields a fundamental approximation

$$E(z) \approx \frac{C}{(1-z)^2} \exp\left(-2p_{\otimes} \int_0^z \frac{A(w)}{1-p_1 w} dw\right) \times \left(2 + \int_0^z G(z) \exp\left(2p_{\otimes} \int_0^{\zeta} \frac{A(w)}{1-p_1 w} dw\right) d\zeta\right)$$

as $z \rightarrow 1$, for a certain constant $C > 0$ and a bounded function $G(z)$.

This means that to study the asymptotic behaviour for $E(z)$ we require quite precise asymptotics regarding $A(z)$ near $z = 1$. In particular, we need to be able to integrate the approximation, and obtain a good approximation after taking the exponential. Thus we need not only an asymptotic equivalent for $A(z)$ as $z \rightarrow 1$, but also a remainder term. The integration involving $A(z)$ is dealt with by the Singular Integration Theorem [9, Thm VI.9].

Analysis of $A(z)$ around its dominant singularity. First we turn the Riccati equation (2) into a linear second order ODE that is homogeneous by a classical change of the unknown function $p_{\otimes} A(z) = v'(z)/v(z)$. We analyze the new function $v(z)$ by the Frobenius method [1, pp.181-182] to obtain a local form of $v(z)$ around the singularity $z = 1$. The conclusion

⁵ We adapt it for our case. In fact $\frac{2}{z}$ is not defined at $z = 0$, where we give our initial condition precisely.

can be found in Proposition 9, which shows that we have 3 regimes for $A(z)$ depending on whether p_{\otimes} is less, equal, or greater than $1/2$. As a by-product, the Transfer Theorem implies (see Theorem 10) that γ_n tends to 0 for $p_{\otimes} \leq 1/2$ and to the constant $\gamma_{\infty} > 0$ when $p_{\otimes} > 1/2$. The detailed analysis is explained in Section 4.

Analysis of $E(z)$ around its dominant singularity. We follow the cases of Proposition 9, which already gives the threshold $1/2$. Then there is an extra threshold coming from the term

$$2 + \int_0^z G(z) \exp\left(2p_{\otimes} \int_0^{\zeta} \frac{A(w)}{1-p_1 w} dw\right) d\zeta$$

in the estimate for $E(z)$. This new threshold corresponds exactly to the point where the integral goes from being convergent to divergent as $z \rightarrow 1$.

For example, when $p_{\otimes} < 1/2$, Proposition 9 yields that the integral $\int_0^z \frac{A(w)}{1-p_1 w} dw$ converges as $z \rightarrow 1$. From our approximation for $E(z)$ we see that actually $E(z) \sim \lambda(1-z)^{-2}$ for a certain constant $\lambda > 0$. By the Transfer Theorem this implies that $e_n \sim \lambda \cdot n$ as $n \rightarrow \infty$.

4 Fully reducible trees

In this section, we study the probability of being fully reducible $\gamma_n = \Pr_n\{\sigma(T) = \mathcal{P}\}$. This is motivated by the fact that γ_n intervenes in the recurrence for the expected value e_n of the size of a random *BST*-like tree after reduction, see Section 3. We recall that we have the following recurrence for $(\gamma_n)_{n \geq 1}$: $\gamma_{n+1} = p_{\otimes} \cdot \frac{1}{n-1} \sum_{k=1}^{n-1} (\gamma_k + \gamma_{n-k} - \gamma_k \gamma_{n-k})$ for all $n \geq s$.

4.1 Generating function and its Riccati equation

As announced, we study γ_n by looking at its generating function $A(z) = \sum_{n=0}^{\infty} \gamma_n z^n$. Note that its radius of convergence is at least 1 because the coefficients γ_n belong to $[0, 1]$. The following proposition shows that it is exactly 1.

► **Proposition 5.** *The radius of convergence of $A(z)$ is exactly 1.*

Proof. We work by contradiction. Suppose $\sum_k \gamma_k$ was convergent. The inequality $\gamma_k + \gamma_{n-k} - \gamma_k \gamma_{n-k} \geq \gamma_k$, valid for all k , implies $\gamma_n \geq \frac{p_{\otimes}}{n-1} \sum_{k=1}^{n-1} \gamma_k = \Omega(1/n)$ from the recurrence in Eq (1). This is absurd because of the divergence of the Harmonic sums. ◀

We recall the Riccati differential equation (2) satisfied by $A(z)$:

$$A'(z) = (s-2)\gamma_s z^{s-1} + \left(\frac{2}{z} + 2p_{\otimes} \frac{z}{1-z}\right) A(z) - p_{\otimes} \cdot (A(z))^2.$$

Consider now the function⁶ $v(z) = \exp(p_{\otimes} \int_0^z A(w) dw)$, which satisfies $A(z) = 1/p_{\otimes} \cdot v'(z)/v(z)$. This is a classical transformation to turn any Riccati equation into a linear ODE of order two. For our case we obtain

$$v''(z) = p_{\otimes} \cdot (s-2)\gamma_s z^{s-1} v(z) + \left(\frac{2}{z} + 2p_{\otimes} \frac{z}{1-z}\right) v'(z). \tag{3}$$

The function $v(z)$ is analytic on the disk $|z| < 1$ as $A(z)$ is analytic there.

⁶ Here \int_0^z means that we integrate on the segment from 0 to z on the complex plane.

The domain of analyticity and the local behaviour of solutions of linear ODE are well understood [1, 16, 18]. We exploit this now to show in Proposition 6 that $A(z)$ is actually analytic on the larger domain $\Omega = \mathbb{C} \setminus [1, \infty)$. Later on (see Prop. 9) we will also use the local form of $v(z)$ to obtain asymptotic equivalents for $A(z)$ around its singularity $z = 1$, which are needed to apply the Transfer Theorem.

► **Proposition 6.** *The power series $A(z)$, seen as an analytic function, can be extended analytically to every point of the domain $\Omega = \mathbb{C} \setminus [1, \infty)$. In particular, $z = 1$ is the only singularity on the circle $|z| = 1$.*

Proof sketch. We already know that $v(z)$ is analytic on the disk $|z| < 1$. Then we use [18, Theorem 2.2, p.3] repeatedly and conclude with the uniqueness of analytic continuation. ◀

4.2 Asymptotics for the fully reducible trees

We can now derive the asymptotic behaviour of $v(z)$, where we recall that $v(z)$ satisfies $v''(x) - \left(\frac{2}{x} + 2p_{\otimes} \frac{x}{1-x}\right) v'(x) - p_{\otimes} \cdot (s-2)\gamma_s x^{s-1} v(x) = 0$, a linear equation of order 2, with non-constant coefficients. We analyze the asymptotics of the solutions close to the singularity by using the Frobenius method (see [1, pp.181-182]). For this we introduce some related notation.

► **Definition 7.** *Consider the homogeneous linear ODE of order two $y''(x) + d_1(x)y'(x) + d_2(x)y(x) = 0$, where $d_1(x)$ and $d_2(x)$ are meromorphic on a star-shaped domain $\tilde{\Omega}$.*

A point $\zeta \in \tilde{\Omega}$ is said to be a regular singularity for the ODE, if it is a singularity of either $d_1(x)$ or $d_2(x)$, or maybe both, and the limits $\delta_j := \lim_{z \rightarrow \zeta} (z - \zeta)^j d_j(z)$, exist and are finite for $j = 1$ and 2. In this case, we define the indicial polynomial $I(\theta)$ at the regular singularity $z = \zeta$ by $I(\theta) = \theta(\theta - 1) + \delta_1\theta + \delta_2$.

The following theorem explains how the indicial polynomial leads to the asymptotics of the solutions⁷:

► **Theorem 8** ([1, Thm 6.14–15]). *Consider the homogeneous linear ODE of order two $y''(x) + d_1(x)y'(x) + d_2(x)y(x) = 0$, where $d_1(x)$ and $d_2(x)$ are meromorphic on a star-shaped domain $\tilde{\Omega}$, and ζ a regular singularity for the given ODE.*

- *If the two roots θ_1 and θ_2 of the indicial polynomial associated to ζ do not differ by an integer (including 0 for double roots), then, in a slit neighbourhood of ζ inside $\tilde{\Omega}$, every solution $y(x)$ is of the form $c_1(\zeta - z)^{\theta_1} H_1(\zeta - z) + c_2(\zeta - z)^{\theta_2} H_2(\zeta - z)$, where $c_1, c_2 \in \mathbb{C}$, and $H_1(z), H_2(z)$ are analytic at $z = 0$ and $H_1(0) \neq 0, H_2(0) \neq 0$.*
- *If the indicial polynomial has a double root θ_0 , then in a slit neighbourhood of ζ inside $\tilde{\Omega}$, every solution $y(x)$ is of the form $(z - \zeta)^{\theta_0} (c_1 H_1(z - \zeta) + c_2 \log(z - \zeta) H_2(z - \zeta))$, where $c_1, c_2 \in \mathbb{C}$, and $H_1(z), H_2(z)$ are analytic at $z = 0$ and $H_1(0) \neq 0, H_2(0) \neq 0$.*

Using this theorem, we directly derive the local behaviour of $v(z)$ and $v'(z)$ around $z = 1$. Now we are ready to obtain the local expansion for $A(z) = \frac{1}{p_{\otimes}} v'(z)/v(z)$, around the singularity $z = 1$, and we prove the following proposition:

► **Proposition 9.** *The ordinary generating function $A(z)$ for $(\gamma_n)_{n \geq 1}$ satisfies the following asymptotic expansions as $z \rightarrow 1$ over Ω*

⁷ The reference uses $|\zeta - z|$ to avoid restricting the domain; here we can use $(\zeta - z)$ because we chose a determination of $\log(1 - z)$.

- For $p_{\otimes} > \frac{1}{2}$, $A(z) = \frac{\gamma_{\infty}}{1-z} + O((1-z)^{2p_{\otimes}-2})$,
- For $p_{\otimes} = \frac{1}{2}$, $A(z) = \frac{2}{1-z} \left(\log \left(\frac{1}{1-z} \right) \right)^{-1} + O \left(\frac{1}{1-z} \left(\log \left(\frac{1}{1-z} \right) \right)^{-2} \right)$
- For $p_{\otimes} < \frac{1}{2}$, $A(z) \sim \frac{D}{(1-z)^{2p_{\otimes}}}$,

where we recall that $\gamma_{\infty} := (2p_{\otimes} - 1)/p_{\otimes}$ and $D > 0$ is a constant depending on p_{\otimes} and s .

As a side product of this proposition, we can apply the Transfer Theorem and show that γ_n indeed converges:

► **Theorem 10.** *The probability γ_n of being fully reducible tends to the constant $\gamma_{\infty} := (2p_{\otimes} - 1)/p_{\otimes}$ for $p_{\otimes} > \frac{1}{2}$ and to zero otherwise. More precisely, for $p_{\otimes} = \frac{1}{2}$ we have $\gamma_n \sim \frac{2}{\log n}$, while for $p_{\otimes} < \frac{1}{2}$, $\gamma_n \sim D \cdot n^{2p_{\otimes}-1}/\Gamma(2p_{\otimes})$, where D is the constant from Prop. 9.*

► **Remark 11.** A different approach for the case $p_{\otimes} < 1/2$ yields the value of the constant for the asymptotics, $D = e^{-2p_{\otimes}} \cdot \left((s-2)\gamma_s \int_0^1 t^{s-3}(1-t)^{2p_{\otimes}} e^{2p_{\otimes}t} dt - p_{\otimes} \int_0^1 (A(t))^2 (1-t)^{2p_{\otimes}} e^{2p_{\otimes}t} dt \right)$. Furthermore, the first term in the parenthesis yields a simple upper-bound.

5 Main result: expected values

This section is devoted to the sketch of the proof of the main theorem:

► **Theorem 12.** *If the probability p_I of unary operators is not zero, then the expected size e_n of a random *BST*-like tree-expression of size n after the bottom-up reduction using an absorbing pattern for the binary operator \otimes satisfies, for some positive constants c_1, \dots, c_4 :*

- if $p_{\otimes} < 1/2$, then $e_n \sim c_4 n$;
- if $p_{\otimes} = 1/2$, then $e_n \sim c_3 n \log(n)^{-2/(1-p_I)}$;
- if $p_{\otimes} > \frac{1}{2}$ and $4p_{\otimes} < 3 - p_I$, then $e_n \sim c_2 n^{1 - \frac{4p_{\otimes}-2}{1-p_I}}$;
- if $4p_{\otimes} = 3 - p_I$, then $e_n \sim c_1 \log(n)$;
- if $4p_{\otimes} > 3 - p_I$, then $e_n \rightarrow e_{\infty}$, where e_{∞} is some positive constant.

Thus we perform a precise study of the generating function $E(z)$ of the expected size e_n . Solving the differential equation satisfied by $E(z)$, we obtain the following as $z \rightarrow 1$

$$E(z) \sim 1 + (1 - p_I)^{2/p_I - 1} K(z)^{-1} \left(2 + \int_0^z G(w)K(w)dw \right) \times (1 - z)^{-2},$$

where $G(z) \rightarrow G(1) > 0$ as $z \rightarrow 1$ and $K(z) := \exp \left(2p_{\otimes} \int_0^z \frac{A(w)}{1-p_I w} dw \right)$.

Then, to obtain the asymptotic estimates we need for applying the Transfer Theorem to $E(z)$, we have to study $K(z)$ and the integral $\int_0^z G(w)K(w)dw$. We remark that the behaviour of the latter is determined roughly by the behaviour of $\int_0^z K(w)dw$. Indeed, if one integral converges, so does the other, and similarly for the divergence. Moreover, when the integral diverges as $z \rightarrow 1$ we also have $\int_0^z G(w)K(w)dw \sim G(1) \int_0^z K(w)dw$.

The asymptotics for $K(z)$ are obtained by the singular integration (see [9, Theorem VI.9]) of the asymptotics of $A(z)$.

► **Example 13.** Consider the case $p_{\otimes} > \frac{1}{2}$. Proposition 9 tells us that $A(z) = \frac{\gamma_{\infty}}{1-z} + O((1-z)^{2p_{\otimes}-2})$. Thus we also have $\frac{A(w)}{1-p_I w} = \frac{\gamma_{\infty}/(1-p_I)}{1-w} + O((1-w)^{2p_{\otimes}-2})$ as $w \rightarrow 1$. Singular integration gives $2p_{\otimes} \int_0^z \frac{A(w)}{1-p_I w} dw = \frac{2p_{\otimes}\gamma_{\infty}}{1-p_I} \log \left(\frac{1}{1-z} \right) + c_0 + O((1-z)^{2p_{\otimes}-1})$ for a certain constant c_0 . As the remainder O -term tends to 0, we conclude $K(z) \sim C_K \times (1-z)^{-2p_{\otimes} \frac{\gamma_{\infty}}{1-p_I}}$. We remark that $\frac{2p_{\otimes}\gamma_{\infty}}{1-p_I} = \frac{4p_{\otimes}-2}{1-p_I}$.

Singular integration yields the following estimates for $J(z) := \int_0^z G(w)K(w)dw$:

► **Lemma 14.** *The function $J(z)$ satisfies the following asymptotics as $z \rightarrow 1$ on Ω :*

- if $4p_{\otimes} > 3 - p_{\text{I}}$, then $J(z) \sim C_J \times (1 - z)^{1 - \frac{4p_{\otimes} - 2}{1 - p_{\text{I}}}}$, with $C_J > 0$
- if $4p_{\otimes} = 3 - p_{\text{I}}$ then $J(z) \sim C_J \times \log\left(\frac{1}{1-z}\right)$, with $C_J > 0$
- if $4p_{\otimes} < 3 - p_{\text{I}}$ then $J(z) \sim C_J$, with $2 + C_J > 0$

where C_J is a constant depending on $p_{\text{I}}, p_{\otimes}, s$.

The proof of this lemma proceeds by discussing whether the integral $J(z)$ is convergent or divergent. Notice for example that $\int_0^z K(w)dw$ diverges for $4p_{\otimes} - 2 \geq 1 - p_{\text{I}}$ due to the estimate given at the end of Example 13. This gives the second threshold $p_{\otimes} = (3 - p_{\text{I}})/4$.

This new threshold $p_{\otimes} = (3 - p_{\text{I}})/4$, along with the previous $p_{\otimes} = \frac{1}{2}$ for the behaviour of $A(z)$, determine the 5 cases in the discussion in Theorem 12.

6 Conclusion

In this article we have seen that random BST-like tree-expressions have a rich range of behaviors with respect to the simple reduction linked to an absorbing pattern. This situation contrasts with the case of uniform random tree-expressions [11, 12] where it was previously shown that the expected value of the size after reduction is $O(1)$.

From a theoretical point of view, the existence of two thresholds is interesting in itself, this leads to a variety of different regimes for the simplifications using a simple rule. There are natural extensions of this paper to widen the result:

- Refine the result for small patterns: this will only improve the multiplicative constants of Theorem 12, not change the order of magnitude of the size of the reduced tree.
- Allow for operators of arity more than 2, as BST-like distribution can naturally be extended to handle such operators. This introduces technical difficulties, but our first attempts at addressing this extension indicate similar results (with different thresholds).
- Allow for more involved specifications, using grammar-like rules. This can be used, for instance, to prevent two consecutive stars in a regular expression. Such specifications were studied for the uniform distribution [12], and require dealing with system of equations instead of just one equation.

However, going back to our initial motivation of analyzing the soundness of random benchmarking, the main continuation of this work would be to mix several simplification procedures. The first step would be to allow several absorbing patterns for different operators together (this was done for a very specific distribution on \wedge/\vee -formulas in [3]). Going even further, we could focus on the simplification procedure of an existing tool and extensively study it using the techniques we developed in this article, for instance, tools like Spot for random LTL-formulas (see Algorithm 1).

To conclude, the message of this paper is that, contrarily to the uniform distribution, a BST-like distribution might be a relevant way to sample interesting random expressions in a practical framework. However, one should be very careful when tuning the parameters, i.e. the probability of the operators, as it may quickly lead to a degenerated case.

References

- 1 T.M. Apostol. *Calculus. Vol. II: Multi-variable Calculus and Linear Algebra, with Applications to Differential Equations and Probability*. Blaisdell international textbook series. Xerox College Publ., 1969.
- 2 Nicolas Broutin and Cécile Mailler. And/or trees: A local limit point of view. *Random Struct. Algorithms*, 53(1):15–58, 2018. doi:10.1002/rsa.20758.
- 3 Brigitte Chauvin, Danièle Gardy, and Cécile Mailler. The growing tree distribution on boolean functions. In *2011 Proceedings of the Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*, pages 45–56, 2011. doi:10.1137/1.9781611973013.5.
- 4 Marco Daniele, Fausto Giunchiglia, and Moshe Y. Vardi. Improved automata generation for linear temporal logic. In Nicolas Halbwachs and Doron Peled, editors, *Computer Aided Verification*, pages 249–260, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.
- 5 Luc Devroye. A note on the height of binary search trees. *J. ACM*, 33(3):489–498, May 1986. doi:10.1145/5925.5930.
- 6 Alexandre Duret-Lutz, Alexandre Lewkowicz, Amaury Fauchille, Thibaud Michaud, Etienne Renault, and Laurent Xu. Spot 2.0 – a framework for LTL and ω -automata manipulation. In *Proceedings of the 14th International Symposium on Automated Technology for Verification and Analysis (ATVA’16)*, volume 9938 of *Lecture Notes in Computer Science*, pages 122–129. Springer, October 2016. doi:10.1007/978-3-319-46520-3_8.
- 7 Philippe Flajolet, Xavier Gourdon, and Conrado Martinez. Patterns in random binary search trees. *Random Struct. Algorithms*, 11(3):223–244, 1997. doi:10.1002/(SICI)1098-2418(199710)11:3<223::AID-RSA2>3.0.CO;2-2.
- 8 Philippe Flajolet and Andrew M. Odlyzko. The average height of binary trees and other simple trees. *J. Comput. Syst. Sci.*, 25(2):171–213, 1982. doi:10.1016/0022-0000(82)90004-6.
- 9 Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.
- 10 Antoine Genitrini, Bernhard Gittenberger, Veronika Kraus, and Cécile Mailler. Associative and commutative tree representations for boolean functions. *Theoretical Computer Science*, 570:70–101, 2015. doi:10.1016/j.tcs.2014.12.025.
- 11 Florent Koechlin, Cyril Nicaud, and Pablo Rotondo. Uniform random expressions lack expressivity. In Peter Rossmanith, Pinar Heggeres, and Joost-Pieter Katoen, editors, *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, August 26-30, 2019, Aachen, Germany*, volume 138 of *LIPICs*, pages 51:1–51:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.
- 12 Florent Koechlin, Cyril Nicaud, and Pablo Rotondo. On the degeneracy of random expressions specified by systems of combinatorial equations. In Natasa Jonoska and Dmytro Savchuk, editors, *Developments in Language Theory - 24th International Conference, DLT 2020, Tampa, FL, USA, May 11-15, 2020, Proceedings*, volume 12086 of *Lecture Notes in Computer Science*, pages 164–177. Springer, 2020.
- 13 Cyril Nicaud. On the Average Size of Glushkov’s Automata. In Adrian-Horia Dediu, Armand-Mihai Ionescu, and Carlos Martín-Vide, editors, *Language and Automata Theory and Applications, Third International Conference, LATA 2009, Tarragona, Spain, April 2-8, 2009. Proceedings*, volume 5457 of *Lecture Notes in Computer Science*, pages 626–637. Springer, 2009.
- 14 Cyril Nicaud, Carine Pivoteau, and Benoît Razet. Average Analysis of Glushkov Automata under a *BST*-Like Model. In Kamal Lodaya and Meena Mahajan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2010)*, volume 8 of *Leibniz International Proceedings in Informatics (LIPICs)*, pages 388–399, Dagstuhl, Germany, 2010. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPICs.FSTTCS.2010.388.

- 15 Carine Pivoteau, Bruno Salvy, and Michèle Soria. Algorithms for combinatorial structures: Well-founded systems and newton iterations. *J. Comb. Theory, Ser. A*, 119(8):1711–1773, 2012. doi:10.1016/j.jcta.2012.05.007.
- 16 Richard P. Stanley. Differentiably finite power series. *Eur. J. Comb.*, 1(2):175–188, 1980. doi:10.1016/S0195-6698(80)80051-5.
- 17 Heikki Tauriainen. Automated testing of Büchi automata translators for linear temporal logic. Research Report A66, Helsinki University of Technology, Laboratory for Theoretical Computer Science, Espoo, Finland, December 2000.
- 18 W. Wasow. *Asymptotic Expansions for Ordinary Differential Equations*. Dover Books on Mathematics. Dover Publications, 2018. URL: <https://books.google.fr/books?id=NQNKDwAAQBAJ>.