Analysis of an efficient reduction algorithm for random regular expressions based on universality detection

Florent Koechlin and Pablo Rotondo

LIGM, Univ Gustave Eiffel, CNRS, ENPC, name.surname@u-pem.fr

Abstract. In this article we study a very simple linear reduction algorithm that is specific to regular expressions. It aims to detect, in a bottom-up fashion, universal subtrees in regular expressions trees, and replace them by the smallest equivalent to Σ^*. Of course, this does not detect every universal subtree, as the universality problem is PSPACE-complete. However, we prove that for the uniform random tree model, this simple algorithm detects a large proportion of universal trees. Furthermore, we prove that on average this algorithm reduces uniform regular expressions to a constant size that is very small, and that can be computed efficiently. For example, for two letters the limit expected size is ~ 77.8. Our theoretical constants are backed-up by the experimental evidence. This confirms the phenomena reported in [13], and further it completely discards the usefulness of the uniform distribution on regular expressions.

1 Introduction

Regular languages are ubiquitous in computer science. The natural way to specify these languages, when it comes to programming, is through regular expressions. Thus there are many algorithms taking regular expressions as inputs. A notable case is the compilation of regular expressions into automata. There exist several constructions of automata from regular expressions; for instance the Thompson construction, the Glushkov position automaton, the partial derivative automaton and the prefix automaton [1,2,16]. Faced with this choice, it is natural to wonder which one performs better in practice. Average case analysis seeks to give an answer by setting up a probabilistic model, hoping that this reflects real life inputs well. In this context, a natural choice for the model is the uniform distribution on the inputs. This distribution has two advantages: it maximizes the entropy, and it is often susceptible to theoretical analysis.

Regular expressions are represented naturally as expression trees, see Figure 1. The uniform distribution on the associated trees has been used with success in the literature to study the complexity of automata constructions [5]. However, this model has been recently put into question by the work in [13]. It is shown that a uniform regular expression tree of size n over a two-letter alphabet is expected to be equivalent to a tree of constant size $3,624,217$, as $n \to \infty$. The equivalent tree can be computed in a bottom-up fashion in linear time. Therefore,

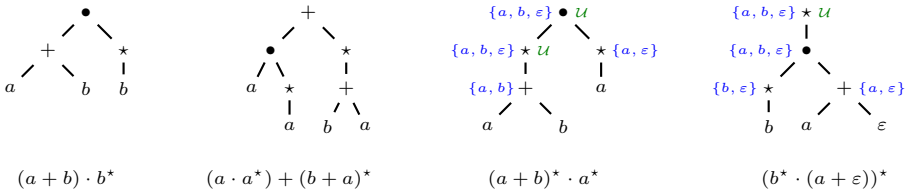

Fig. 1. Four regular expression trees and their associated formulas. The last three are universal, i.e., they recognize every word on the alphabet $\{a, b\}$. Our linear simplification algorithm (see Figure 2) will detect their universality. For example, the last two expression trees are annotated with the subset (in blue) of symbols a, b, ε recognized and (in green) whether the expression associated to the node is detected to be universal by our algorithm.

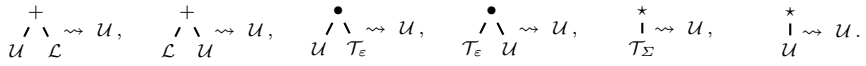

Fig. 2. The bottom-up set of reduction rules. Here $\mathcal{U}$ is a special tree representing those identified as universal. Then $\mathcal{L}$ denotes any tree, $\mathcal{T}_\Sigma$ denotes any tree recognizing every letter of Σ, and $\mathcal{T}_\varepsilon$ denotes the class of trees recognizing the empty word ε. Note for example that $\mathcal{U} \in \mathcal{T}_\varepsilon \cap \mathcal{T}_\Sigma$. Hence the last rule is redundant, but written for better understanding.

the asymptotic average case analysis is doomed to be trivial once the expression has been reduced (in time $O(n)$).

Even if the smaller equivalent expression obtained in [13] is constant size on average, its size ($\approx 3.6 \cdot 10^6$) is not small enough to exclude its usefulness in random generation of regular expressions. But the size of the output expression is likely overestimated due to the fact that the model considered in [13] is very general; it applies to a large range of expression specified by expression trees (e.g., logical formulas, artihmetic expressions, ...). Furthermore, several works on boolean expressions trees [6,11] have shown that considering the finer details of the semantic rules for a concrete case might lead to much stronger results (and a smaller constant).

In this work, we study the application of a simplification algorithm that is specific to regular expression trees and exploits their particular semantic properties[1]. The procedure seeks to recognize when a tree represents a universal expression (i.e., recognizing every possible word). The algorithm must be efficient (linear time) as it is intended to be used in a pre-processing step to simplify the expressions. In Figure 2 we show the schema of the reduction rules applied by recognizing universality. The reduction works bottom-up. We record for each tree the subset of $\Sigma \cup \{\varepsilon\}$ that is recognized by the expression, depicted in blue in Figure 1. In addition, the algorithm tries to detect if the expression is universal

The complete code is detailed in Appendix A.

[1] However, this idea may be adaptable to other similar bottom-up procedures.

(i.e. accepts every word on the alphabet). As the universality problem for regular expressions is PSPACE-complete [15], this detection (in linear time) is bound to be incomplete. The reduction consists in replacing the identified universal subtrees of the input expression by a fixed tree $\mathcal{U}$ representing the class of universal trees. This tree $\mathcal{U}$ may be taken to be, for example, a universal one of minimal size (e.g., $(a+b)^\star$ for two letters), or a completely new alphabet symbol.

It could be said that this is the most natural and simplest algorithm to detect in linear time as many universal subtrees as possible. As mentioned previously, this algorithm does not detect all universal trees; consider for instance the tree associated with $L = \Sigma \cdot \Sigma^\star + \varepsilon$. The algorithm does not realize that $\Sigma \cdot \Sigma^\star$ is only missing ε to be universal. Note that the procedure detects the universality of the three trees on the right of Figure 1, which in contrast cannot be reduced at all by the absorbing-pattern procedure given in [13].

Our main contribution is showing, without a shadow of doubt, that the uniform model is not apt for practical use. We already know due to [13] that the size after reduction is bounded by a huge constant[2]. Note that the convergence of the expected size is not immediate in our case because the reduction is significantly different. In this article we prove that our algorithm yields a limit expected size which is significantly smaller. We give a general method to compute this limit to any arbitrary precision, that works for any alphabet size, and is efficient. The limits for $k = |\Sigma|$ from two to five are shown in Table 1, where we have taken $\mathcal{U}$ to be a minimal universal tree of size $2k$. For instance, for an alphabet on two symbols ($k = 2$) this yields a constant ~ 77.797, which is prohibitively small. This is confirmed by our experiments (see Fig. 3) for a wide range of sizes. Further, the experiments suggest that the limit might even be close to being an upper-bound.

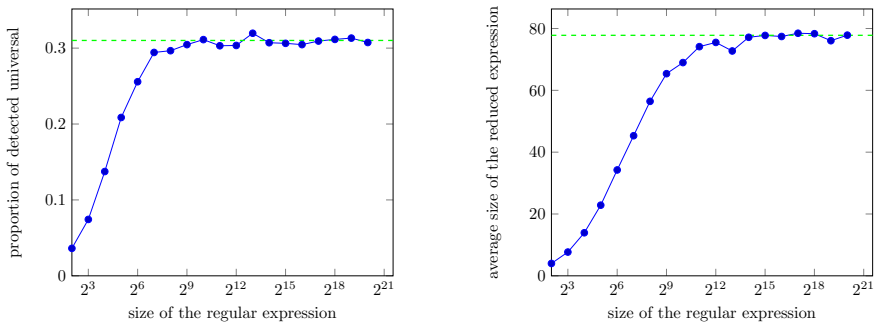

Fig. 3. The proportion of universal expressions detected by the algorithm, and the average size after reduction, observed experimentally[3] on regular expressions on two letters, with 10 , 000 samples for each size. The plots are in log-scale for the input sizes. The theoretical limits are marked by green (dashed) lines.

[2] Their algorithm is a sub-reduction of ours in the case of regular expressions.
[3] For the simulations, the uniform trees were sampled using the algorithm in [7].

Theorem. *Consider the simple variety of tree expressions encoding regular expressions for an alphabet of fixed size $|\Sigma| = k$, and σ the linear-time simplification induced by the rules in Figure 2. Then the expected size of a uniform random expression of size n after simplification tends to a constant as n tends to infinity.*

$\lvert\Sigma\rvert$	2	3	4	5
$\lim \mathbb{E}_n[\lvert\sigma(T)\rvert]$	77.79724…	495.59151…	2 518.20513…	11 694.43727…

Table 1.

Moreover, Table 1 shows the limits, for alphabets of size k up to 5, computed to five exact digits after the decimal point.

As a second result of independent interest, we provide bounds for the proportion of universal expression trees. In Proposition 10, we show that there is a high proportion of expression trees which represent universal expressions. In particular, the proportion is asymptotically comprised between 0.31 and 0.46 for an alphabet on two letters. See Fig. 3 for the lower bound. Thus the fact that the reduced trees have a limit for their expected size can be thought of, intuitively, as a consequence of the preponderance of universal expressions.

We conclude the introduction by giving a plan of the article. Section 2 introduces the definitions and the basic techniques from Analytic Combinatorics [10] that we will employ. In particular the reduction algorithm, and the main generating functions. Next, in Section 3 we study the generating functions of the combinatorial classes associated with the algorithm, and the size after reduction. Theorem 1 gives a recursive system of the combinatorial classes. Using marking techniques on the system, we describe the reduced size in terms of a bivariate generating function. Then Theorem 2 proves that these classes have a limit probability, and we conclude the section with Theorem 3 which describes the expected values. Finally, in Section 4 we show how to compute the limits (probability and expectation) efficiently to arbitrary precision[4]. This involves a rewriting of the system in a simpler form (see Sec. 4.1). The procedure[5] works for any value of $k = |\Sigma|$.

The proofs are provided in the appendices.

The proofs are either sketched or completely omitted in this extended abstract.

2 Model and definitions

2.1 Expression trees: definitions and counting

We introduce the family of trees representing regular expressions. The trees considered throughout this article are rooted and planar: $\overset{op}{\wedge}_{T_1\ T_2}$ and $\overset{op}{\wedge}_{T_2\ T_1}$ do not represent the same tree.

Definition 1. *Given a finite alphabet $\mathcal{A} = \{a_1, \ldots, a_k\}$, we define the class of regular expression trees $\mathcal{L}_\mathcal{R} = \mathcal{L}_\mathcal{R}(\mathcal{A})$ on $\mathcal{A}$ inductively from the equation*

$$\mathcal{L}_\mathcal{R} = a_1 + \ldots + a_k + \varepsilon + \overset{\star}{|}_{\mathcal{L}_\mathcal{R}} + \overset{\bullet}{\wedge}_{\mathcal{L}_\mathcal{R}\ \mathcal{L}_\mathcal{R}} + \overset{+}{\wedge}_{\mathcal{L}_\mathcal{R}\ \mathcal{L}_\mathcal{R}}. \tag{2.1}$$

[4] We could in fact compute them exactly, but their exact expression is not readable

[5] The code is provided in Sage and Maple at `https://igm.univ-mlv.fr/~koechlin/csr_reduction_universality/`

The size $|T|$ of an expression tree $T \in \mathcal{L}_{\mathcal{R}}$ is defined to be its number of nodes. In particular the leaves $a_1, \ldots, a_k$ and ε have size 1.

For $n \in \mathbb{N}$, we note $\mathcal{L}_n$ the set of regular expressions of size n. In our model we fix n the size of tree, and draw $T \in \mathcal{L}_n$ uniformly. The probability of picking a particular $T \in \mathcal{L}_n$ is then $1/|\mathcal{L}_n|$. Now we show how to obtain the asymptotics for $|\mathcal{L}_n|$, and how to use this to obtain the probability of recognizing ε.

Formal generating series. In order to count the trees, and obtain asymptotics, we make use of the framework of Analytic Combinatorics [10]. In particular, we deal with ordinary generating functions (OGFs for short).

The ordinary generating function $L(z)$ associated with $\mathcal{L}_{\mathcal{R}}$ is defined as the formal power series $L(z) := \sum_{T \in \mathcal{L}_{\mathcal{R}}} z^{|T|} = \sum_{n \geq 0} \ell_n z^n$ where $\ell_n = |\mathcal{L}_n|$.

The equation defining the class of expression trees $\mathcal{L}_{\mathcal{R}}$ translates into a functional equation for its ordinary generating function:

$$L(z) = (k+1)z + zL(z) + 2z(L(z))^2 \,. \tag{2.2}$$

This translation from the inductive equation, Eq (2.1), to the functional equation, Eq (2.2), comes from general principles of Analytic Combinatorics: the disjoint union of two classes is associated to the sum of their OGFs, and the cartesian product of two classes translates into the product of their OGFs [10].

Transfer Theorem. This symbolic translation is the first of the two steps employed in the Analytic Combinatorics study of the coefficients. For the second, we consider the OGF $C(z) = \sum c_n z^n$ as a function on the complex plane $\mathbb{C}$. The behaviour of $C(z)$ at its dominant (closest to the origin) singularities translates into asymptotics for its coefficients $c_n \geq 0$. We use the notation $[z^n]C(z) := c_n$, for the coefficients of $C(z)$. *Pringsheim's Theorem* [10, Theorem IV.6] implies that the radius of convergence $\rho = \rho_C > 0$ of $C(z)$ is a dominant singularity. Then the celebrated *Transfer Theorem* [10, Ch VI.3] states that, under certain analytic conditions, if ρ is the only singularity on the circle $|z| = \rho$ and we have the local estimate $C(z) \sim_{z \to \rho} \lambda(1 - z/\rho)^{-\beta}$, with $\beta \notin \{0, -1, -2, \ldots\}$, around $z = \rho$, then we have the asymptotics $[z^n]C(z) \sim_{n \to \infty} \lambda \rho^{-n} n^{\beta - 1}/\Gamma(\beta)$, where Γ is Euler's gamma-function, for the coefficients of $C(z)$.

Asymptotics for the number of trees. The Equation (2.2) is quadratic in $L(z)$. Thus we can solve for the generating function, obtaining $L(z) = \left(1 - z - \sqrt{\Delta(z)}\right)/(4z)$ with $\Delta(z) := -(8k+7)\,z^2 - 2\,z + 1$, which is the only combinatorially sound solution. Then $L(z)$ presents a false singularity at $z = 0$, and a unique dominant singularity ρ at the root of $\Delta(z)$ that is closest to the origin. The value of the singularity ρ and the value of $L(\rho)$ are characterized by

$$L(\rho) = \sqrt{\frac{1+k}{2}}\,, \qquad \rho = \frac{1}{1 + 4L(\rho)}\,.$$

Since ρ is a simple root of $\Delta(z)$, we derive that $L(z) = h_L - g_L\sqrt{1-z/\rho} + O\left(\left|1-\frac{z}{\rho}\right|\right)$ as $z \to \rho$, where $h_L = L(\rho)$, and g_L can be obtained by differentiation, namely $g_L = 2\rho \lim_{z\to\rho} L'(z)\cdot\sqrt{1-z/\rho}$. Thus the Transfer Theorem[6] yields

$$\ell_n = [z^n]L(z) \sim -g_L\rho^{-n}n^{-3/2}/\Gamma(-1/2) = g_L\rho^{-n}n^{-3/2}/(2\sqrt{\pi})\,.$$

Probability of recognizing ε. We present a basic class of regular expressions that intervene crucially in our work: the tree expressions recognizing the empty word.

The class $\mathcal{T}_\varepsilon$ of tree expressions recognizing ε can be characterized inductively by the following equation where the decomposition into sum of classes is disjoint:

$$\mathcal{T}_\varepsilon = \varepsilon + \overset{\star}{|}_{\mathcal{L}_\mathcal{R}} + \overset{\bullet}{/\!\backslash}_{\mathcal{T}_\varepsilon\ \mathcal{T}_\varepsilon} + \overset{+}{/\!\backslash}_{\mathcal{T}_\varepsilon\ \mathcal{L}_\mathcal{R}} + \overset{+}{/\!\backslash}_{\mathcal{L}_\mathcal{R}\ \mathcal{T}_\varepsilon\ \mathcal{T}_\varepsilon}\,.$$

Thus we obtain the OGF $T_\varepsilon(z)$ by the principles of Analytic Combinatorics:

$$T_\varepsilon(z) = z + zL(z) + 2zL(z)T_\varepsilon(z)\,. \tag{2.3}$$

Solving the linear equation, we can verify that $T_\varepsilon(z) = h_{T_\varepsilon} - g_{T_\varepsilon}\sqrt{1-z/\rho} + O\,(1-z/\rho)$ as $z \to \rho$, with a constant $g_{T_\varepsilon} \neq 0$. Thus the number of tree expressions of size n recognizing ε is asymptotically

$$[z^n]T_\varepsilon(z) \sim g_{T_\varepsilon}\rho^{-n}n^{-3/2}/(2\sqrt{\pi})\,.$$

For the proof, see Annex B.

Normalizing by ℓ_n, we obtain the following proposition:

Proposition 1. *The probability of a random uniform tree of size n recognizing ε converges, as $n \to \infty$, to a positive constant $g_{T_\varepsilon}/g_L = \frac{\sqrt{2k+2}+3/2}{k+\sqrt{2k+2}+3/2}$.*

2.2 The reduction process

We consider $\mathcal{R}$ a particular subclass of trees recognizing every word of $\Sigma^\star$:

Definition 2. *The class $\mathcal{R}$ is defined inductively:*

- *if T recognizes every letter of the alphabet, then $\overset{\star}{|}_T \in \mathcal{R}$;*
- *if at least one of T_1 or T_2 belongs to $\mathcal{R}$, then $\overset{+}{/\!\backslash}_{T_1\ T_2} \in \mathcal{R}$;*
- *if $T_1 \in \mathcal{R}$ and $T_2 \in \mathcal{T}_\varepsilon$, then $\overset{\bullet}{/\!\backslash}_{T_1\ T_2} \in \mathcal{R}$ and $\overset{\bullet}{/\!\backslash}_{T_2\ T_1} \in \mathcal{R}$.*

In particular if $T \in \mathcal{R}$, then $\overset{\star}{|}_T \in \mathcal{R}$.

Note that $\mathcal{R}$ is the class of subtrees reduced to $\mathcal{U}$ by the algorithm. As announced in the introduction, the tree associated with $\Sigma\cdot\Sigma^\star + \varepsilon$ is not in $\mathcal{R}$ whereas it is universal.

[6] Strictly speaking, we should deal with the remainder term.

Recognizing letters. In order to decide whether a tree belongs to $\mathcal{R}$, by using the definition bottom-up, we must be able to decide (also bottom-up) whether a tree T recognizes a given letter a. This is done as follows:

- if $|T| = 1$, then T recognizes a if and only if $T = a$,
- if the root of T is either $\star$ or $+$, then T recognizes a if and only if one of its children recognizes the letter a,
- if the root of T is $\bullet$, then T recognizes a if and only if one of its children recognizes the letter a and the other one recognizes ε.

Example 1. If T_1 recognizes ε and a, while T_2 recognizes ε and b, then $\overset{\bullet}{\underset{T_1}{/}\underset{T_2}{\backslash}}$ recognizes a, b and ε.

Definition 3 (Reduction algorithm σ). *Given a tree T, and a fixed tree $\mathcal{U}$ representing $\Sigma^\star$, we produce the reduced tree $\sigma_{\mathcal{U}}(T)$ as follows. We begin bottom-up from the leaves, and we keep track of whether the current tree: (1) recognizes each letter of Σ, (2) recognizes ε, (3) is in $\mathcal{R}$. The veracity of all of these predicates is determined bottom-up as described above. Whenever a subtree is in $\mathcal{R}$, we substitute it by $\mathcal{U}$. When $\mathcal{U}$ is clear by context, we simply write $\sigma(T)$.*

2.3 Generating functions with additional parameters

The generating function $L(z)$ only counts the number ℓ_n of trees of a given size n. To keep track of the reduced size of the trees at the same time, we introduce a new "marking" variable u and consider the *bivariate* generating function

$$L(z, u) = \sum_{T \in \mathcal{L}_{\mathcal{R}}} z^{|T|} u^{|\sigma(T)|} . \tag{2.4}$$

Then the expected size of a tree of size n after reduction can be expressed by:

$$\mathbb{E}_n[|\sigma(T)|] = \frac{[z^n]\partial_u L(z, u)\big|_{u=1}}{[z^n]L(z)} . \tag{2.5}$$

We need information about $L(z, u)$ to evaluate the numerator in Eq. (2.5). In order to find a suitable expression for $L(z, u)$, we will split $\mathcal{L}$ into several subclasses. These subclasses correspond to the different stages in building an element from $\mathcal{R}$, and take the reduction into account (see Section 3.1).

3 Analytic characterization of the limit

The objective of this section is to show that the expected size after reduction $\sigma(T)$ converges as the size n of the random tree tends to infinity. To do this, we study the analytic properties of $\partial_u L(z, u)|_{u=1}$, the derivative in u of the bivariate generating function defined in Eq. (2.4). The analytic properties presented here will also be needed in Section 4, where we show how to exploit them to obtain an efficient and high-precision procedure to compute the limit of the expectation.

3.1 Combinatorial system for the reduction

Following the construction of the class $\mathcal{R}$, we introduce the following notation: for every subset of letters $X \subseteq \Sigma$, $\mathcal{T}_{X,\varepsilon}$ denotes the set of trees that recognize the empty word, every letter in X, but no letter in $\Sigma \backslash X$. Similarly we denote by $\mathcal{T}_{X,\overline{\varepsilon}}$ the set of tree expression that recognize every letter in X, but no letter in $\Sigma \backslash X$, nor the empty word ε.

Example 2. For instance, $\mathcal{T}_{\{a\},\varepsilon}$ contains the trees for $a^\star$ and $(a \cdot (b^\star) + \varepsilon)$, but does not contain the tree a.

Theorem 1. *The combinatorial classes $(\mathcal{T}_{X,\varepsilon})_{X\subseteq\Sigma}$ and $(\mathcal{T}_{X,\overline{\varepsilon}})_{X\subseteq\Sigma}$ satisfy the inductive definition:*

$$\begin{aligned}
\mathcal{T}_{X,\varepsilon} &= \varepsilon \mathbf{1}_{X=\emptyset} + \overset{\star}{|}_{\mathcal{T}_{X,\varepsilon}} + \overset{\star}{|}_{\mathcal{T}_{X,\overline{\varepsilon}}} + \sum_{(S,S'):S\cup S'=X} \mathcal{T}_{S,\varepsilon} \overset{\bullet}{\wedge} \mathcal{T}_{S',\varepsilon} \\
&\quad + \sum_{(S,S'):S\cup S'=X} \mathcal{T}_{S,\varepsilon} \overset{+}{\wedge} \mathcal{T}_{S',\varepsilon} + \sum_{(S,S'):S\cup S'=X} \mathcal{T}_{S,\varepsilon} \overset{+}{\wedge} \mathcal{T}_{S',\overline{\varepsilon}} + \sum_{(S,S'):S\cup S'=X} \mathcal{T}_{S,\overline{\varepsilon}} \overset{+}{\wedge} \mathcal{T}_{S',\varepsilon}, \\
\mathcal{T}_{X,\overline{\varepsilon}} &= X \mathbf{1}_{|X|=1} + \sum_{S\subseteq\Sigma} \mathcal{T}_{X,\overline{\varepsilon}} \overset{\bullet}{\wedge} \mathcal{T}_{S,\varepsilon} + \sum_{S\subseteq\Sigma} \mathcal{T}_{S,\varepsilon} \overset{\bullet}{\wedge} \mathcal{T}_{X,\overline{\varepsilon}} + \mathbf{1}_{X=\emptyset} \sum_{S,S'\subseteq\Sigma} \mathcal{T}_{S,\overline{\varepsilon}} \overset{\bullet}{\wedge} \mathcal{T}_{S',\overline{\varepsilon}} \\
&\quad + \sum_{(S,S'):S\cup S'=X} \mathcal{T}_{S,\overline{\varepsilon}} \overset{+}{\wedge} \mathcal{T}_{S',\overline{\varepsilon}},
\end{aligned}$$

where the union $S \cup S'$ need not be disjoint, but the sums $+$ are all disjoint.

Proof (Sketch). This is just an exhaustive enumeration of every possible case for a tree to recognize any set of letters and the empty word. We however have to be careful to produce an unambiguous specification.

Adding $\mathcal{R}$ to the system. The class of fully reducible trees $\mathcal{R}$ satisfies the equation:

$$\mathcal{R} = \overset{\star}{|}_{\mathcal{T}_{\Sigma,\overline{\varepsilon}}} + \overset{\star}{|}_{\mathcal{T}_{\Sigma,\varepsilon}} + \mathcal{R} \overset{+}{\wedge} \mathcal{L} + \mathcal{L}\backslash\mathcal{R} \overset{+}{\wedge} \mathcal{R} + \mathcal{R} \overset{\bullet}{\wedge} \mathcal{T}_{\varepsilon} + \mathcal{T}_{\varepsilon}\backslash\mathcal{R} \overset{\bullet}{\wedge} \mathcal{R}, \tag{3.1}$$

We want to add this equation to our system. For the terms to remain positive we introduce the class $\mathcal{T}_G := \mathcal{T}_{\Sigma,\varepsilon} \setminus \mathcal{R}$, namely, the class of trees recognizing every letter and the empty word, that are not fully reducible. Then we have the disjoint sum $\mathcal{T}_{\Sigma,\varepsilon} = \mathcal{R} + \mathcal{T}_G$. Hence, in Eq. (3.1), we can expand $\mathcal{L} = \sum_{X\subsetneq\Sigma} \mathcal{T}_{X,\varepsilon} + \mathcal{T}_G + \mathcal{R} + \sum_{X\subseteq\Sigma} \mathcal{T}_{X,\overline{\varepsilon}}$ and $\mathcal{T}_\varepsilon = \sum_{X\subsetneq\Sigma} \mathcal{T}_{X,\varepsilon} + \mathcal{R} + \mathcal{T}_G$. In particular this gives $\mathcal{L} \setminus \mathcal{R} = \mathcal{T}_G + \sum_{X\subsetneq\Sigma} \mathcal{T}_{X,\varepsilon} + \sum_{X\subseteq\Sigma} \mathcal{T}_{X,\overline{\varepsilon}}$, and similarly for $\mathcal{T}_\varepsilon \setminus \mathcal{R}$.

For $\mathcal{T}_G$ we have a similar equation, which we derive from expanding $\mathcal{T}_{\Sigma,\varepsilon} = \mathcal{R} + \mathcal{T}_G$ in the equation for $\mathcal{T}_{\Sigma,\varepsilon}$ and eliminating the terms involving $\mathcal{R}$. In particular there are no trees in $\mathcal{T}_G$ having $\star$ as root, and those having $\bullet$ as root are

$$\sum_{\substack{S\subsetneq\Sigma, S'\subsetneq\Sigma: \\ S\cup S'=\Sigma}} \mathcal{T}_{S,\varepsilon} \overset{\bullet}{\wedge} \mathcal{T}_{S',\varepsilon} + \sum_{S\subsetneq\Sigma} \mathcal{T}_G \overset{\bullet}{\wedge} \mathcal{T}_{S,\varepsilon} + \mathcal{T}_G \overset{\bullet}{\wedge} \mathcal{T}_G + \sum_{S\subsetneq\Sigma} \mathcal{T}_{S,\varepsilon} \overset{\bullet}{\wedge} \mathcal{T}_G,$$

as we must prohibit the subtrees from being in $\mathcal{R}$. For $+$ to be the root, we must prohibit having one subtree that recognizes ε and the other in $\mathcal{R}$. This is easily calculated but a bit long to write out in full.

The full specification of $\mathcal{T}_G$ can be found in Annex C

3.2 Generating functions and probability of full reduction

From the combinatorial system of Theorem 1, in $(\mathcal{T}_{X,\overline{\varepsilon}}, \mathcal{T}_{X,\varepsilon})_{X\subseteq\Sigma}$, we have introduced a new one in $(\mathcal{R}, \mathcal{T}_G, \mathcal{T}_{\Sigma,\overline{\varepsilon}}, (\mathcal{T}_{X,\overline{\varepsilon}}, \mathcal{T}_{X,\varepsilon})_{X\subsetneq\Sigma})$. In this section we look at the system of their generating functions and prove several basic properties.

We denote by $y_{X,\varepsilon}(z)$ (resp. $y_{X,\overline{\varepsilon}}(z)$) the generating series of $\mathcal{T}_{X,\varepsilon}$ (resp. $\mathcal{T}_{X,\overline{\varepsilon}}$). Similarly, we denote by $R(z)$ and $y_G(z)$ the OGFs of $\mathcal{R}$ and $\mathcal{T}_G$. Note in particular that $y_{\Sigma,\varepsilon}(z) = R(z) + y_G(z)$. Henceforth we will write as a column vector

$$\boldsymbol{y}(z) = \left[R(z), y_G(z), y_{\Sigma,\overline{\varepsilon}}(z), (y_{X,\overline{\varepsilon}}(z), y_{X,\varepsilon}(z))_{X\subsetneq\Sigma}\right].$$

The full proofs for this subsection can be found in Annex D

Proposition 2. *The vector $\boldsymbol{y}(z)$ satisfies a vectorial system under the form:*

$$\boldsymbol{y}(z) = \boldsymbol{\Phi}(z; \boldsymbol{y}(z)) \tag{3.2}$$

where each component of $\boldsymbol{\Phi}(z; \boldsymbol{y})$ is a polynomial of degree 2 in $\boldsymbol{y}$, of degree 1 in z, such that $\boldsymbol{\Phi}(0; \boldsymbol{y}) = \mathbf{0}$.

Remark 1. If $X, Y \subseteq \Sigma$ have the same number of letters $|X| = |Y|$, then $y_{X,\varepsilon}(z) = y_{Y,\varepsilon}(z)$ and $y_{X,\overline{\varepsilon}}(z) = y_{Y,\overline{\varepsilon}}(z)$. This follows from picking any isomorphism permuting the letters of Σ and mapping X to Y. Thus in reality we may rewrite the system in just $1 + 2 \times (k+1)$ equations where $k = |\Sigma|$, rather than the exponential $1 + 2^{k+1}$ we would have by considering every subset.

Proposition 3. *Every coordinate of the solution $\boldsymbol{y}(z)$ of the system has ρ as a unique dominant singularity, such that near $z = \rho$:*

$$\boldsymbol{y}(z) = \boldsymbol{h}(z) - \boldsymbol{g}(z)\sqrt{1 - z/\rho} \tag{3.3}$$

where $\boldsymbol{h}(z)$ and $\boldsymbol{g}(z)$ are two vectors of analytic functions in a neighbourhood of $z = \rho$. Besides, every coordinate of $\boldsymbol{g}(\rho)$ is strictly positive.

Proof (sketch). We prove that the system (3.2) is strongly connected, so that we can apply Drmota's theorem [8]. The common singularity is already known since it must coincide with the singularity of $L(z)$.

Theorem 2. *The probability that a random tree T of size n belongs to a class $\mathcal{C}$ of the extended combinatorial system tends to a positive constant $g_{\mathcal{C}}(\rho)/g_L(\rho)$ as $n \to \infty$. In particular, the limit probability of a full reduction ($\mathcal{C} = \mathcal{R}$) is positive.*

Proof (Sketch). Proposition 3 implies that $R(z) = h_R(z) - g_R(z)\sqrt{1 - z/\rho}$ around $z = \rho$, with $g_R(\rho) \neq 0$. Since there is no other singularity on the circle $|z| = \rho$, we obtain the asymptotics from the Transfer Theorem and the proof follows as that of the probability of recognizing ε.

3.3 Extended system for the expected value

To deal with the expected value, as explained in Section 2.3, we introduce a new variable u which will "mark" the size of the reduced expression. We extend each generating function to two variables $\boldsymbol{y}(z,u)$. It is immediate to see that $R(z,u) = u^{|\mathcal{U}|}R(z)$. Note that for the other classes, the root always remains after the reduction. Hence we almost have for them the same equations in two variables than in one variable, with an additional factor u to count the root in the size of the reduced tree. We summarize this discussion in the following proposition.

Proposition 4. *Let us write $\boldsymbol{y} = (R, \tilde{\boldsymbol{y}})$, and $\boldsymbol{\Phi} = (\Phi_R, \tilde{\boldsymbol{\Phi}})$. The vector of bivariate generating functions $\tilde{\boldsymbol{y}}(z,u)$ satisfies the vectorial system:*

$$\tilde{\boldsymbol{y}}(z,u) = \tilde{\boldsymbol{\Phi}}(zu; u^p R(z), \tilde{\boldsymbol{y}}(z,u))$$

where $\boldsymbol{\Phi} = (\Phi_R, \tilde{\boldsymbol{\Phi}})$ is defined in Eq. 3.2, and $p := |\mathcal{U}|$.

Notice that $L(z,u) = u^p R(z) + (1,\dots,1)\cdot\tilde{\boldsymbol{y}}(z,u)$. Following Eq (2.5), we need to differentiate $L(z,u)$ on u then set $u=1$ to find the expected value. For notation convenience, we will write $Q\tilde{\boldsymbol{y}}(z) := \partial_u \tilde{\boldsymbol{y}}(z,u)\big|_{u=1}$.

Proposition 5. *The vector $Q\tilde{\boldsymbol{y}}(z) = \partial_u \tilde{\boldsymbol{y}}(z,u)\big|_{u=1}$ satisfies the linear system:*

$$(\mathtt{Id} - \mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](z; R(z), \tilde{\boldsymbol{y}}(z)))Q\tilde{\boldsymbol{y}}(z) = \tilde{\boldsymbol{\Phi}}(z; R(z), \tilde{\boldsymbol{y}}(z)) + p\partial_R \tilde{\boldsymbol{\Phi}}(z; R(z), \tilde{\boldsymbol{y}}(z))R(z)$$

Proof. It is straightforward by differentiating Proposition 4. Note that $z\partial_z \tilde{\boldsymbol{\Phi}} = \tilde{\boldsymbol{\Phi}}$.

Hence we can show that $Q\tilde{\boldsymbol{y}}(z)$ has a dominant square-root singularity at $z=\rho$:

Proposition 6. *Every coordinate of the vector $Q\tilde{\boldsymbol{y}}(z) = \partial_u \tilde{\boldsymbol{y}}(z,u)\big|_{u=1}$ has a unique dominant singularity at $z=\rho$. Further, near $z=\rho$ we may write:*

$$Q\tilde{\boldsymbol{y}}(z) = \boldsymbol{h}_{Q\tilde{\boldsymbol{y}}}(z) - \boldsymbol{g}_{Q\tilde{\boldsymbol{y}}}(z)\sqrt{1 - z/\rho}$$

where $\boldsymbol{h}_{Q\tilde{\boldsymbol{y}}}(z)$ and $\boldsymbol{g}_{Q\tilde{\boldsymbol{y}}}(z)$ are two vectors of analytic functions in a neighbourhood of $z=\rho$, such that every coordinate of $\boldsymbol{g}_{Q\tilde{\boldsymbol{y}}}(\rho)$ is strictly positive.

For the full proofs for this subsection, see Annex E.

Proof (Sketch). We prove that we can inverse the matrix in Proposition 5, and use Proposition 3 to prove that the solution has the right form.

Using Eq (2.5) and the Transfer Theorem, we can finally conclude:

Theorem 3 (Limit of the expected size). *Consider the simple variety of tree expressions encoding regular expressions for an alphabet of fixed size $|\Sigma| = k$, and the linear-time simplification algorithm σ. Then the expected size of a uniform random expression of size n after simplification by σ tends to a constant as n tends to infinity:*

$$\lim_{n\to+\infty} \mathbb{E}_n[|\sigma(T)|] = \frac{|\mathcal{U}| g_R(\rho) + \|\boldsymbol{g}_{Q\tilde{\boldsymbol{y}}}(\rho)\|_1}{g_L(\rho)} \tag{3.4}$$

where $\|(v_1,\dots,v_s)\|_1 = |v_1| + \dots + |v_s|$.

Remark 2 (size of $\mathcal{U}$). A natural value for the size of $\mathcal{U}$ is $|\mathcal{U}| = 2k$ if we represent universality by any minimal unary-binary tree for $\Sigma^\star$, or $|\mathcal{U}| = 1$ if we use a special symbol. We remark that one must be careful when changing $\mathcal{U}$ as the vector $(g_R(z), \boldsymbol{g}_{Q\tilde{\boldsymbol{y}}}(z))$, evaluated in Eq. (3.4), also depends on $|\mathcal{U}|$.

4 Practical computation of the limit: a numerical study

The main goal of this section is to give an effective procedure to compute the constant in Theorem 3, for any size of the alphabet.

According to Eq (3.4), we need to compute $g_R(\rho)$ and $\boldsymbol{g}_{Q\tilde{\boldsymbol{y}}}(\rho)$. We notice that for any analytic function $w(z)$ under the form $w(z) = h(z) - g(z)\sqrt{1-z/\rho}$, then $w'(z) = O(1) + \frac{g(\rho)}{2\rho\sqrt{1-z/\rho}}$ for $z \sim \rho$. Hence $g(\rho) = \lim_{z\to\rho} 2\rho\, w'(z)\sqrt{1-z/\rho}$.

Differentiating in z the system in Prop. 5 leads to the following proposition:

see Annex F for the complete proofs of this section

Proposition 7. *The vector $g_{Q\tilde{\boldsymbol{y}}}(\rho)$ satisfies the equation:*

$$g_{Q\tilde{\boldsymbol{y}}}(\rho) = \Big(\mathtt{Id} - \mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](\rho; \boldsymbol{y}(\rho))\Big)^{-1} \times \boldsymbol{K}_{\boldsymbol{\Phi}}(\rho; \boldsymbol{y}(\rho), g_{\boldsymbol{y}}(\rho), Q\tilde{\boldsymbol{y}}(\rho))$$

where $\boldsymbol{K}_{\boldsymbol{\Phi}}(z; \boldsymbol{y}, \boldsymbol{g}, \boldsymbol{h})$ depends on the derivatives of $\boldsymbol{\Phi}$, $p = |\mathcal{U}|$, and it is polynomial in its input vectors.

Hence we need to compute $\boldsymbol{y}(\rho), g_{\boldsymbol{y}}(\rho)$ and $Q\tilde{\boldsymbol{y}}(\rho)$. This is done in three steps:

1. To compute $\boldsymbol{y}(\rho)$, we rewrite the system in a "triangular form". This is done by exploiting the known functions $L(z)$ and $T_\varepsilon(z)$. Then $\boldsymbol{y}(\rho)$ can be effectively computed by dynamic programming. See Section 4.1 for details.
2. Then $\boldsymbol{g}_{\boldsymbol{y}}(\rho)$ is computed by solving a linear system, as it is an eigenvector of the matrix $\mathtt{Jac}_{\boldsymbol{y}}[\boldsymbol{\Phi}](\rho, \boldsymbol{y}(\rho))$. This is explained in Section 4.2.
3. Finally, setting $z = \rho$ in Prop. 5 yields a simple matrix formula for $Q\tilde{\boldsymbol{y}}(\rho)$. The inverse of the matrix is well-defined, as shown in the proof of Prop. 6.

4.1 Triangular form of the system

Factorizing terms in the equations for the combinatorial classes for $\mathcal{T}_{X,\overline{\varepsilon}}$ in Theorem 1, the combinatorial classes $\mathcal{T}_\varepsilon$ and $\mathcal{T}_{\overline{\varepsilon}} := \mathcal{L}_{\mathcal{R}} \backslash \mathcal{T}_\varepsilon$ turn up. This is summarized in the following proposition.

Proposition 8. *The combinatorial class $(\mathcal{T}_{X,\overline{\varepsilon}})_{X\subseteq\Sigma}$ satisfies the inductive definition:*

$$\mathcal{T}_{X,\overline{\varepsilon}} = X\mathbf{1}_{|X|=1} + \mathcal{T}_{X,\overline{\varepsilon}} \overset{\bullet}{\wedge} \mathcal{T}_\varepsilon + \mathcal{T}_\varepsilon \overset{\bullet}{\wedge} \mathcal{T}_{X,\overline{\varepsilon}} + \mathbf{1}_{X=\emptyset}\, \mathcal{T}_{\overline{\varepsilon}} \overset{\bullet}{\wedge} \mathcal{T}_{\overline{\varepsilon}} + \sum_{(S,S'):S\cup S'=X} \mathcal{T}_{S,\overline{\varepsilon}} \overset{+}{\wedge} \mathcal{T}_{S',\overline{\varepsilon}}.$$

This yields a new triangular system for the generating functions: we obtain a quadratic equation in $y_{X,\overline{\varepsilon}}(z)$ whose coefficients involve only $y_{S,\overline{\varepsilon}}$ with $S \subsetneq X$, similarly, we have a quadratic equation for $y_{X,\varepsilon}(z)$ whose coefficients involve only $y_{S,\overline{\varepsilon}}$ with $S \subsetneq X$ and also $y_{X,\overline{\varepsilon}}$. We can then compute fast the numerical solution[7] for $\boldsymbol{y}(\rho)$.

[7] In fact, we can solve the system in $\boldsymbol{y}(z)$ exactly. However the closed-form solutions become huge; for instance, for $\Sigma = \{a, b\}$:
$y_{\Sigma,\overline{\varepsilon}}(z) = \frac{1}{4z}\Big(-\sqrt{\Delta(z)} + 2\sqrt{(2z+2)\sqrt{\Delta(z)} - 6z^2 + 2} - \sqrt{(2z+2)\sqrt{\Delta(z)} + 10z^2 + 2} - z - 1\Big)$.

Algorithm to compute $\boldsymbol{y}(\rho)$. First we compute $L(\rho)$, $T_\varepsilon(\rho)$ and $T_{\bar{\varepsilon}}(\rho)$ as explained in Section 2.1. Then, given the triangular system, we compute the values of $y_{X,\varepsilon}(\rho)$ and $y_{X,\bar{\varepsilon}}(\rho)$ for each $X \subseteq \Sigma$ by dynamic programming. Each step requires simple operations (sums, products, and a single square-root) on previously computed values. Finally, $R(\rho)$ is computed from Eq. (3.1), while $y_G(\rho) = y_{\Sigma,\varepsilon}(\rho) - R(\rho)$.

4.2 Limit probabilities as an eigenvector

The vector $\boldsymbol{g}_{\boldsymbol{y}}(\rho)$ is characterized in terms of a linear system of equations.

Proposition 9. *The coefficients* $\boldsymbol{g}_{\boldsymbol{y}}(\rho)$ *constitute an eigenvector for* $\lambda = 1$ *for the Jacobian matrix* $\mathtt{Jac}_{\boldsymbol{y}}[\boldsymbol{\Phi}](\rho; \boldsymbol{y}(\rho))$ *at* $z = \rho$*, namely*

$$\mathtt{Jac}_{\boldsymbol{y}}[\boldsymbol{\Phi}](\rho; \boldsymbol{y}(\rho)) \cdot \boldsymbol{g}_{\boldsymbol{y}}(\rho) = \boldsymbol{g}_{\boldsymbol{y}}(\rho)\,.$$

Furthermore, the eigenspace associated to $\lambda = 1$ *has dimension* 1 *and* $\boldsymbol{g}_{\boldsymbol{y}}(\rho)$ *is characterized as the only eigenvector satisfying* $\|\boldsymbol{g}_{\boldsymbol{y}}(\rho)\|_1 = g_L(\rho)$.

Probabilities of each class. In particular, for two letters we obtain $\lim_n \Pr_n(\mathcal{R}) \doteq 0.310122\ldots$, while $\lim_n \Pr_n(\mathcal{T}_{\Sigma,\varepsilon}) \doteq 0.457051\ldots$. These constitute bounds for the proportion of universal expressions. We summarize in the following:

Proposition 10. *For all* n *large enough, the proportion* $\Pr_n(\mathtt{univ.})$ *of trees representing universal expressions over a* k*-letter alphabet belongs to the intervals:*

k	2	3	4	5
`interval`	$(0.31, 0.46)$	$(0.13, 0.27)$	$(0.062, 0.15)$	$(0.028, 0.077)$

5 Conclusion

We have provided a simple linear algorithm reducing a random regular expression to an equivalent one that on average has a small constant size. This shows that the uniform tree model is most definitely flawed when it comes to producing random regular expressions, as it produces very limited languages.

An interesting aspect to highlight is the combinatorial system characterizing the reduction process (see Sec. 3.1), in particular its simplicity, and the fact that it allows for efficient computation and (big) exact solutions.

The simplification process relies on detecting universality. Our study reveals that universality is abundant in the random uniform model. Moreover, the proportion of universal trees is comprised in a small range (see Proposition 10). We could refine the detection algorithm by considering slight improvements. The goal is to, for instance, recognize the universality of $L = \Sigma \cdot \Sigma^\star + \varepsilon$. An idea is to consider not only whether $\Sigma^\star$ is recognized but also $a \cdot \Sigma^\star$ and $b \cdot \Sigma^\star$. More generally, one can consider for a given depth k, whether the words in $\Sigma^{\leq k}$ are recognized, and also the sets $w \cdot \Sigma^\star$ for a prefix free set of $w \in \Sigma^{\leq k}$. For large k this should lead to better upper and lower bounds for the asymptotic probability of a tree being universal. Hopefully these bounds will coalesce as $k \to \infty$, which would then prove the existence of the limit probability for universality.

References

1. Allauzen, C., Mohri, M.: A unified construction of the glushkov, follow, and antimirov automata. In: Královič, R., Urzyczyn, P. (eds.) Mathematical Foundations of Computer Science 2006. pp. 110–121. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
2. Antimirov, V.: Partial derivatives of regular expressions and finite automata constructions. In: Mayr, E.W., Puech, C. (eds.) STACS 95. pp. 455–466. Springer Berlin Heidelberg, Berlin, Heidelberg (1995)
3. Apostol, T.: Calculus. Vol. II: Multi-variable Calculus and Linear Algebra, with Applications to Differential Equations and Probability. Blaisdell international textbook series, Xerox College Publ. (1969)
4. Bell, J.P., Burris, S., Yeats, K.A.: Characteristic points of recursive systems. Electr. J. Comb. **17**(1) (2010)
5. Broda, S., Machiavelo, A., Moreira, N., Reis, R.: On the average size of glushkov and partial derivative automata. Int. J. Found. Comput. Sci. **23**(5), 969–984 (2012). https://doi.org/10.1142/S0129054112400400, https://doi.org/10.1142/S0129054112400400
6. Chauvin, B., Gardy, D., Mailler, C.: The growing tree distribution on Boolean functions., pp. 45–56. https://doi.org/10.1137/1.9781611973013.5, https://epubs.siam.org/doi/abs/10.1137/1.9781611973013.5
7. Devroye, L.: Simulating size-constrained galton-watson trees. SIAM J. Comput. **41**(1), 1–11 (2012). https://doi.org/10.1137/090766632, https://doi.org/10.1137/090766632
8. Drmota, M.: Systems of functional equations. Random Struct. Algorithms **10**(1-2), 103–124 (1997)
9. Drmota, M.: Random Trees: An Interplay Between Combinatorics and Probability. Springer Publishing Company, Incorporated, 1st edn. (2009)
10. Flajolet, P., Sedgewick, R.: Analytic Combinatorics. Cambridge University Press (2009)
11. Gardy, D.: Random boolean expressions. In: DMTCS Proceedings vol. AF, Computational Logic and Applications (CLA '05). pp. 1–36. Discrete Mathematics & Theoretical Computer Science, Episciences. org (2005)
12. Godsil, C.D., Royle, G.F.: Algebraic Graph Theory. Graduate texts in mathematics, Springer (2001)
13. Koechlin, F., Nicaud, C., Rotondo, P.: Uniform random expressions lack expressivity. In: Rossmanith, P., Heggernes, P., Katoen, J. (eds.) 44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, August 26-30, 2019, Aachen, Germany. LIPIcs, vol. 138, pp. 51:1–51:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2019)
14. Koechlin, F., Nicaud, C., Rotondo, P.: On the degeneracy of random expressions specified by systems of combinatorial equations. In: Jonoska, N., Savchuk, D. (eds.) Developments in Language Theory - 24th International Conference, DLT 2020, Tampa, FL, USA, May 11-15, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12086, pp. 164–177. Springer (2020)
15. Meyer, A.R., Stockmeyer, L.J.: The equivalence problem for regular expressions with squaring requires exponential space. In: Proceedings of the 13th Annual Symposium on Switching and Automata Theory (Swat 1972). p. 125–129. SWAT '72, IEEE Computer Society, USA (1972). https://doi.org/10.1109/SWAT.1972.29, https://doi.org/10.1109/SWAT.1972.29

16. Yamamoto, H.: A new finite automaton construction for regular expressions. In: Bensch, S., Freund, R., Otto, F. (eds.) Sixth Workshop on Non-Classical Models for Automata and Applications - NCMA 2014, Kassel, Germany, July 28-29, 2014. Proceedings. books@ocg.at, vol. 304, pp. 249–264. Österreichische Computer Gesellschaft (2014)
17. Yosida, K.: Functional analysis. Classics in Mathematics, Springer-Verlag, Berlin (1995), reprint of the sixth (1980) edition
18. Zedek, M.: Continuity and location of zeros of linear combinations of polynomials. Proceedings of the American Mathematical Society **16**(1), 78–84 (1965)

A Detailed pseudo-code of the reduction algorithm σ

In this section we give the full pseudo-code of our reduction algorithm on the alphabet Σ. The reduction corresponds to the first component of the return value of the extended function `reduce`, which returns the pair

$$\texttt{reduced}(T) = (\sigma(T), S_T)\,,$$

where $S_T \subseteq \{\varepsilon\} \cup \Sigma$ denotes the set of leaves recognized by T.

function reduce(T):
if $|T| = 1$ **then**
  return $(T, \{T\})$;
if $T = \overset{+}{\underset{T_L}{/}\backslash\,\underset{T_R}{}}$ **then**
  $(T'_L, S_L) :=$ reduce(T_L);
  $(T'_R, S_R) :=$ reduce(T_R);
  if $T'_L = \mathcal{U}$ *or* $T'_R = \mathcal{U}$ **then**
    return $(\mathcal{U}, S_L \cup S_R)$;
  return $(\overset{+}{\underset{T'_L}{/}\backslash\,\underset{T'_R}{}}, S_L \cup S_R)$;
else if $T = \overset{\bullet}{\underset{T_L}{/}\backslash\,\underset{T_R}{}}$ **then**
  $(T'_L, S_L) :=$ reduce(T_L);
  $(T'_R, S_R) :=$ reduce(T_R);
  $S := \emptyset$;
  if $\varepsilon \in S_R$ **then**
    if $T'_L = \mathcal{U}$ **then**
      return $(\mathcal{U}, S_L)$;
    $S := S \cup S_L$;
  if $\varepsilon \in S_L$ **then**
    if $T'_R = \mathcal{U}$ **then**
      return $(\mathcal{U}, S_R)$;
    $S := S \cup S_R$;
  return $(\overset{\bullet}{\underset{T'_L}{/}\backslash\,\underset{T'_R}{}}, S)$;
else if $T = \overset{\star}{\underset{T_0}{|}}$ **then**
  $(T', S') :=$ reduce(T_0);
  if $\Sigma \subseteq S'$ **then**
    return $(\mathcal{U}, \{\varepsilon\} \cup \Sigma)$;
  return $(\overset{\star}{\underset{T'}{|}}, \{\varepsilon\} \cup S')$;

B Proofs from Section 2.1

Proposition 1. *The probability of a random uniform tree of size n recognizing ε converges, as $n \to \infty$, to a positive constant $g_{T_\varepsilon}/g_L = \frac{\sqrt{2k+2}+3/2}{k+\sqrt{2k+2}+3/2}$.*

Proof. Recall that $L(z) = h_L - g_L\sqrt{1-z/\rho} + O\left(\left|1-\frac{z}{\rho}\right|\right)$ as $z \to \rho$. Hence plugging this expression into $T_\varepsilon(z) = \frac{z+zL(z)}{1-2zL(z)}$, and using the asymptotic expansion of $(1-2zL(z))^{-1} = (1-2\rho h_L)^{-1}(1 - \frac{2zg_L}{1-2zh_L}\sqrt{1-z/\rho} + O(1-z/\rho))$, we obtain that $g_{T_\varepsilon}/g_L = \frac{4\rho(1+2\rho)}{(1+\rho)^2}$. Plugging in the expression for ρ we get the result. □

C Proofs from Section 3.1

Theorem 1. *The combinatorial classes $(\mathcal{T}_{X,\varepsilon})_{X\subseteq\Sigma}$ and $(\mathcal{T}_{X,\bar{\varepsilon}})_{X\subseteq\Sigma}$ satisfy the inductive definition:*

$$\begin{aligned}
\mathcal{T}_{X,\varepsilon} &= \varepsilon\mathbf{1}_{X=\emptyset} + \overset{\star}{|}_{\mathcal{T}_{X,\varepsilon}} + \overset{\star}{|}_{\mathcal{T}_{X,\bar{\varepsilon}}} + \sum_{(S,S'):S\cup S'=X} \mathcal{T}_{S,\varepsilon} \overset{\bullet}{\wedge} \mathcal{T}_{S',\varepsilon} \\
&\quad + \sum_{(S,S'):S\cup S'=X} \mathcal{T}_{S,\varepsilon} \overset{+}{\wedge} \mathcal{T}_{S',\varepsilon} + \sum_{(S,S'):S\cup S'=X} \mathcal{T}_{S,\varepsilon} \overset{+}{\wedge} \mathcal{T}_{S',\bar{\varepsilon}} + \sum_{(S,S'):S\cup S'=X} \mathcal{T}_{S,\bar{\varepsilon}} \overset{+}{\wedge} \mathcal{T}_{S',\varepsilon}, \\
\mathcal{T}_{X,\bar{\varepsilon}} &= X\mathbf{1}_{|X|=1} + \sum_{S\subseteq\Sigma} \mathcal{T}_{X,\bar{\varepsilon}} \overset{\bullet}{\wedge} \mathcal{T}_{S,\varepsilon} + \sum_{S\subseteq\Sigma} \mathcal{T}_{S,\varepsilon} \overset{\bullet}{\wedge} \mathcal{T}_{X,\bar{\varepsilon}} + \mathbf{1}_{X=\emptyset} \sum_{S,S'\subseteq\Sigma} \mathcal{T}_{S,\bar{\varepsilon}} \overset{\bullet}{\wedge} \mathcal{T}_{S',\bar{\varepsilon}} \\
&\quad + \sum_{(S,S'):S\cup S'=X} \mathcal{T}_{S,\bar{\varepsilon}} \overset{+}{\wedge} \mathcal{T}_{S',\bar{\varepsilon}},
\end{aligned}$$

where the union $S \cup S'$ need not be disjoint, but the sums $+$ are all disjoint.

Proof. Let $X \subseteq \Sigma$, and let us consider $\mathcal{T}_{X,\varepsilon}$, the class of expression trees recognizing every letter in X, the empty word, but no letter in $\Sigma \setminus X$.

A leaf, which is always labelled by an element of $\Sigma \cup \{\varepsilon\}$, is in $\mathcal{T}_{X,\varepsilon}$ if and only if $X = \emptyset$ and the leaf is labelled by ε. Hence the term $\varepsilon\mathbf{1}_{X=\emptyset}$.

The rest of the terms are obtained by considering the root of the trees in $\mathcal{T}_{X,\varepsilon}$:

- a tree $\overset{\star}{|}_{T}$ belongs to $\mathcal{T}_{X,\varepsilon}$ if and only if T recognizes every letter of X and no other one (it does not matter if T recognizes ε or not), i.e. $T \in \mathcal{T}_{X,\varepsilon}$ or $T \in \mathcal{T}_{X,\bar{\varepsilon}}$. Both cases are disjoint, hence the term $\overset{\star}{|}_{\mathcal{T}_{X,\varepsilon}} + \overset{\star}{|}_{\mathcal{T}_{X,\bar{\varepsilon}}}$;
- for a tree $T = \overset{\bullet}{\wedge}_{T_1\ T_2}$ to belong to $\mathcal{T}_{X,\varepsilon}$, it is necessary that both T_1 and T_2 recognize ε, otherwise T would not recognize ε. Then $T_1 \in \mathcal{T}_{S,\varepsilon}$ and $T_2 \in \mathcal{T}_{S',\varepsilon}$ for some $S, S' \subseteq \Sigma$. As ε is recognized by both tree, we have $X = S \cup S'$. Reciprocally the concatenation of $T_1 \in \mathcal{T}_{S,\varepsilon}$ and $T_2 \in \mathcal{T}_{S',\varepsilon}$ for any sets S, S' such that $X = S \cup S'$ belongs to $\mathcal{T}_{X,\varepsilon}$. Noticing that every such pairs (S, S') enumerate disjoint cases, this leads to the term $\sum_{(S,S'):S\cup S'=X} \mathcal{T}_{S,\varepsilon} \overset{\bullet}{\wedge} \mathcal{T}_{S',\varepsilon}$;
- for a tree $T = \overset{+}{\wedge}_{T_1\ T_2}$ to belong to $\mathcal{T}_{X,\varepsilon}$, it is necessary that at least one of the children T_1 or T_2 recognize ε. The disjonction whether it is T_1, T_2 or both leads to the last three terms of the equation for $\mathcal{T}_{X,\varepsilon}$.

Let us consider now the set $\mathcal{T}_{X,\overline{\varepsilon}}$ of expression trees recognizing every letter in X, no letter in $\Sigma \setminus X$, nor the empty word ε.

A leaf, which is always labelled by an element of $\Sigma \cup \{\varepsilon\}$, is in $\mathcal{T}_{X,\overline{\varepsilon}}$ if and only if $X = \{x\}$ and the leaf is labelled by $x \in \Sigma$. Hence the term $X\mathbf{1}_{|X|=1}$. The rest of the terms are obtained by considering the root of the trees in $\mathcal{T}_{X,\overline{\varepsilon}}$:

- a tree having $\star$ as a root always recognizes ε, and hence can never be in $\mathcal{T}_{X,\overline{\varepsilon}}$;
- for a tree $T = \overset{+}{\underset{T_1}{/}\underset{T_2}{\backslash}}$ to belong to $\mathcal{T}_{X,\overline{\varepsilon}}$, it is necessary that none of its children recognize ε. Hence $T_1 \in \mathcal{T}_{S,\overline{\varepsilon}}$ and $T_2 \in \mathcal{T}_{S',\overline{\varepsilon}}$ for some $S, S' \subseteq \Sigma$ such that $S \cup S' = X$. Reciprocally, trees of this form belong to $\mathcal{T}_{X,\overline{\varepsilon}}$, and the partitioning according to the pair S, S' is unambiguous. Hence the term $\sum_{(S,S'):S\cup S'=X} \overset{+}{\underset{\mathcal{T}_{S,\overline{\varepsilon}}}{/}\underset{\mathcal{T}_{S',\overline{\varepsilon}}}{\backslash}}$ in the equation;
- for the case where the root is $\bullet$, we notice that for $T = \overset{\bullet}{\underset{T_1}{/}\underset{T_2}{\backslash}}$ to belong to $\mathcal{T}_{X,\overline{\varepsilon}}$, it is impossible that both T_1 and T_2 recognize ε, otherwise T would recognize ε too. We then consider the three disjoint cases:
  - if $T_1 \in \mathcal{T}_{S',\overline{\varepsilon}}$ and $T_2 \in \mathcal{T}_{S,\varepsilon}$ where $S, S' \subseteq \Sigma$, then as their concatenation belongs to $\mathcal{T}_{X,\overline{\varepsilon}}$, we must have $S' = X$. Reciprocally the concatenation of $T_1 \in \mathcal{T}_{X,\overline{\varepsilon}}$ and $T_2 \in \mathcal{T}_{S,\varepsilon}$ for any $S \subseteq \Sigma$ belongs to $\mathcal{T}_{X,\overline{\varepsilon}}$. Hence the term $\sum_{S\subseteq\Sigma} \overset{\bullet}{\underset{\mathcal{T}_{X,\overline{\varepsilon}}}{/}\underset{\mathcal{T}_{S,\varepsilon}}{\backslash}}$;
  - if $T_1 \in \mathcal{T}_{S,\varepsilon}$ and $T_2 \in \mathcal{T}_{S',\overline{\varepsilon}}$ where $S, S' \subseteq \Sigma$, we have symmetric term;
  - finally if neither T_1 not T_2 recognize ε, that is $T_1 \in \mathcal{T}_{S,\overline{\varepsilon}}$ and $T_2 \in \mathcal{T}_{S',\overline{\varepsilon}}$ for some $S, S' \subseteq \Sigma$, then their concatenation cannot recognize any letter nor ε, so X must be the empty set. Reciprocally, for $X = \emptyset$, the concatenation of any $T_1 \in \mathcal{T}_{S,\overline{\varepsilon}}$ and $T_2 \in \mathcal{T}_{S',\overline{\varepsilon}}$ belongs to $\mathcal{T}_{\emptyset,\overline{\varepsilon}}$. Hence the term $\mathbf{1}_{X=\emptyset} \sum_{S,S'\subseteq\Sigma} \overset{\bullet}{\underset{\mathcal{T}_{S,\overline{\varepsilon}}}{/}\underset{\mathcal{T}_{S',\overline{\varepsilon}}}{\backslash}}$.

□

Equation for $\mathcal{T}_G$. Remember that in the article we split the class $\mathcal{T}_{\Sigma,\varepsilon}$ into two disjoint classes, $\mathcal{R}$ and $\mathcal{T}_G$. We recall that a direct reasoning gives the equation satisfied by $\mathcal{R}$:

$$\mathcal{R} = \overset{\star}{|}_{\mathcal{T}_{\Sigma,\overline{\varepsilon}}} + \overset{\star}{|}_{\mathcal{T}_{\Sigma,\varepsilon}} + \overset{+}{\underset{\mathcal{R}}{/}\underset{\mathcal{L}}{\backslash}} + \overset{+}{\underset{\mathcal{L}\setminus\mathcal{R}}{/}\underset{\mathcal{R}}{\backslash}} + \overset{\bullet}{\underset{\mathcal{R}}{/}\underset{\mathcal{T}_\varepsilon}{\backslash}} + \overset{\bullet}{\underset{\mathcal{T}_\varepsilon\setminus\mathcal{R}}{/}\underset{\mathcal{R}}{\backslash}}, \tag{3.1}$$

However the equation for $\mathcal{T}_G$ is longer to write. In fact it is simply obtained from the equation for $\mathcal{T}_{\Sigma,\varepsilon}$ by removing the trees starting by a $\star$, and replacing every occurrence of $\mathcal{T}_{\Sigma,\varepsilon}$ by $\mathcal{T}_G$. We write here the equation for completeness:

$$\mathcal{T}_G = \sum_{\substack{S\subsetneq\Sigma, S'\subsetneq\Sigma:\\ S\cup S'=\Sigma}} \mathcal{T}_{S,\varepsilon} \overset{\bullet}{\wedge} \mathcal{T}_{S',\varepsilon} + \sum_{S\subsetneq\Sigma} \mathcal{T}_G \overset{\bullet}{\wedge} \mathcal{T}_{S,\varepsilon} + \mathcal{T}_G \overset{\bullet}{\wedge} \mathcal{T}_G + \sum_{S\subsetneq\Sigma} \mathcal{T}_{S,\varepsilon} \overset{\bullet}{\wedge} \mathcal{T}_G$$
$$+ \sum_{\substack{S\subsetneq\Sigma, S'\subsetneq\Sigma:\\ S\cup S'=\Sigma}} \mathcal{T}_{S,\varepsilon} \overset{+}{\wedge} \mathcal{T}_{S',\varepsilon} + \sum_{S\subsetneq\Sigma} \mathcal{T}_G \overset{+}{\wedge} \mathcal{T}_{S,\varepsilon} + \mathcal{T}_G \overset{+}{\wedge} \mathcal{T}_G + \sum_{S\subsetneq\Sigma} \mathcal{T}_{S,\varepsilon} \overset{+}{\wedge} \mathcal{T}_G$$
$$+ \sum_{\substack{S\subsetneq\Sigma, S'\subsetneq\Sigma:\\ S\cup S'=\Sigma}} \mathcal{T}_{S,\varepsilon} \overset{+}{\wedge} \mathcal{T}_{S',\overline{\varepsilon}} + \sum_{S\subsetneq\Sigma} \mathcal{T}_G \overset{+}{\wedge} \mathcal{T}_{S,\varepsilon} + \sum_{\substack{S\subsetneq\Sigma, S'\subsetneq\Sigma:\\ S\cup S'=\Sigma}} \mathcal{T}_{S,\overline{\varepsilon}} \overset{+}{\wedge} \mathcal{T}_{S',\varepsilon} + \sum_{S\subseteq\Sigma} \mathcal{T}_{S,\varepsilon} \overset{+}{\wedge} \mathcal{T}_G$$

D Proofs from Section 3.2

Removing any occurrence of $\mathcal{T}_{\Sigma,\varepsilon}$ from Theorem 1 by replacing it by $\mathcal{R}+\mathcal{T}_G$, we obtain a combinatorial system for the vector of classes

$$(\mathcal{R}, \mathcal{T}_G, \mathcal{T}_{\Sigma,\overline{\varepsilon}}, (\mathcal{T}_{X,\overline{\varepsilon}}, \mathcal{T}_{X,\varepsilon}(z))_{X\subsetneq\Sigma}) .$$

We recall that we denote by $y_{X,\varepsilon}(z)$ (resp. $y_{X,\overline{\varepsilon}}(z)$) the generating series of $\mathcal{T}_{X,\varepsilon}$ (resp. $\mathcal{T}_{X,\overline{\varepsilon}}$). Similarly, we denote by $R(z)$ and $y_G(z)$ the OGFs of $\mathcal{R}$ and $\mathcal{T}_G$. Remember in particular that $y_{\Sigma,\varepsilon}(z) = R(z) + y_G(z)$. We introduce a vector-style notation:

$$\boldsymbol{y}(z) = [R(z), y_G(z), y_{\Sigma,\overline{\varepsilon}}, (y_{X,\overline{\varepsilon}}(z), y_{X,\varepsilon}(z))_{X\subsetneq\Sigma}] .$$

Proposition 2. *The vector $\boldsymbol{y}(z)$ satisfies a vectorial system under the form:*

$$\boldsymbol{y}(z) = \boldsymbol{\Phi}(z; \boldsymbol{y}(z)) \tag{3.2}$$

where each component of $\boldsymbol{\Phi}(z;\boldsymbol{y})$ is a polynomial of degree 2 in $\boldsymbol{y}$, of degree 1 in z, such that $\boldsymbol{\Phi}(0;\boldsymbol{y}) = \mathbf{0}$.

Proof. The translation from the combinatorial description in Theorem 1, enhanced with the equations for $\mathcal{R}$ and $\mathcal{T}_G$, into a system over the OGF is straightforward. For completeness, we provide the equation for $y_{X,\overline{\varepsilon}}(z)$, where $X \subseteq \Sigma$, for which the combinatorial equation:

$$\mathcal{T}_{X,\overline{\varepsilon}} = X\mathbf{1}_{|X|=1} + \sum_{(S,S'):S\cup S'=X} \mathcal{T}_{S,\overline{\varepsilon}} \overset{+}{\wedge} \mathcal{T}_{S',\overline{\varepsilon}}$$
$$+ \sum_{S\subseteq\Sigma} \mathcal{T}_{X,\overline{\varepsilon}} \overset{\bullet}{\wedge} \mathcal{T}_{S,\varepsilon} + \sum_{S\subseteq\Sigma} \mathcal{T}_{S,\varepsilon} \overset{\bullet}{\wedge} \mathcal{T}_{X,\overline{\varepsilon}} + \mathbf{1}_{X=\emptyset} \sum_{S,S'\subseteq\Sigma} \mathcal{T}_{S,\overline{\varepsilon}} \overset{\bullet}{\wedge} \mathcal{T}_{S',\overline{\varepsilon}}$$

translates into the functional equation:

$$y_{X,\overline{\varepsilon}}(z) = z\mathbf{1}_{|X|=1} + z \sum_{(S,S'):S\cup S'=X} y_{S,\overline{\varepsilon}}(z) y_{S',\overline{\varepsilon}}(z)$$
$$+ z \sum_{S\subseteq\Sigma} y_{X,\overline{\varepsilon}}(z) y_{S,\varepsilon}(z) + z \sum_{S\subseteq\Sigma} y_{S,\varepsilon} y_{X,\overline{\varepsilon}}(z) + z\mathbf{1}_{X=\emptyset} \sum_{S,S'\subseteq\Sigma} y_{S,\overline{\varepsilon}}(z) y_{S',\overline{\varepsilon}}(z)$$

It suffices then to replace $y_{\Sigma,\varepsilon}(z)$ by $R(z) + T_G(z)$ to obtain an equation

$$y_{X,\bar{\varepsilon}}(z) = \boldsymbol{\Phi}_{X,\bar{\varepsilon}}(z, \boldsymbol{y}(z))$$

where $\boldsymbol{\Phi}_{X,\bar{\varepsilon}}(z, \boldsymbol{y})$ is a polynomial in z and y, of degree 1 in z, with z in factor, and of degree 2 in $\boldsymbol{y}$.

It is easy to see that the latter property is also satisfied for $\boldsymbol{\Phi}_{X,\varepsilon}, \boldsymbol{\Phi}_R, \boldsymbol{\Phi}_G$. This concludes the proof. □

Now we move on to Prop. 3 which shows that the entries of $\boldsymbol{y}(z)$ satisfy the hypothesis of the Transfer Theorem. More precisely we will show that $\rho = \rho_L$ is the sole ingularity of the entries and that they are of the form $\sim h - g\sqrt{1 - z/\rho}$ when $z \to \rho$. In order to do this there is classical theorem by Drmota [9, Theorem 2.33]. Drmota's Theorem yields the analytical properties of the solutions of a combinatorial system of equations under good conditions that we will verify.

The result of applying Drmota is summarized in the following proposition.

Proposition 3. *Every coordinate of the solution $\boldsymbol{y}(z)$ of the system has ρ as a unique dominant singularity, such that near $z = \rho$:*

$$\boldsymbol{y}(z) = \boldsymbol{h}(z) - \boldsymbol{g}(z)\sqrt{1 - z/\rho} \tag{3.3}$$

where $\boldsymbol{h}(z)$ and $\boldsymbol{g}(z)$ are two vectors of analytic functions in a neighbourhood of $z = \rho$. Besides, every coordinate of $\boldsymbol{g}(\rho)$ is strictly positive.

For the proof, as announced, we use a classical result of Drmota [9, Theorem 2.33] for strongly connected systems of polynomial equations. We need to verify the hypotheses of the theorem, in particular we highlight the following ones which correspond to lemmas below (the rest appear at the end of the proof):

- The underlying directed-graph $G_{\boldsymbol{\Phi}}$ is strongly connected (see definition next), which we prove in Lemma 1.
- The solution generating functions are aperiodic, see Lemma 2.
- Drmota's Theorem tells us that the components have all the same radius of convergence, we prove in Lemma 3 that this coincides with the radius of convergence of L, $\rho = \rho_L$.

To simplify the notations, we will sometimes enumerate the coordinates, and write $\boldsymbol{y} = (y_i)_{i=1\ldots m}$ and $\boldsymbol{\Phi} = (\boldsymbol{\Phi}_i)_{i=1\ldots m}$ in this proof, where $m = 2^{k+1} + 1$.

We denote by $G_{\boldsymbol{\Phi}}$ the dependency graph associated with $\boldsymbol{\Phi}$: it has m nodes, labelled by $y_1 \ldots y_m$, and there is an oriented arc from y_i to y_j if the degree in y_j of the polynomial $\boldsymbol{\Phi}_i(z, \boldsymbol{y})$ is bigger than 1: in other words we have $y_i \to y_j$ whenever $\deg_{y_j}(\Phi) \geq 1$.

Lemma 1. *The graph $G_{\boldsymbol{\Phi}}$ is strongly connected.*

Proof. Let us look at the system from Theorem 1:

- for every $X \subsetneq \Sigma$, $\mathcal{T}_{X,\varepsilon}$ depends on $\mathcal{T}_{\emptyset,\bar{\varepsilon}}$ (by taking $S = \emptyset$ in the last sum). $\mathcal{T}_G$ and $\mathcal{R}$ also depend in their equation on $\mathcal{T}_{\emptyset,\bar{\varepsilon}}$. Hence, for any node $y \in \{y_G, y_R, (y_{X,\varepsilon})_{X \subsetneq \Sigma}\}$, there is an arc from y to the node $y_{\emptyset,\bar{\varepsilon}}$ in $G_{\boldsymbol{\Phi}}$.

– As in the expression of $\boldsymbol{\Phi}_{\emptyset,\overline{\varepsilon}}$ we have the polynomial $z\sum_{S,S'\subseteq\Sigma} y_{S,\overline{\varepsilon}}y_{S',\overline{\varepsilon}}$, there is in $G_{\boldsymbol{\Phi}}$ an arc from $y_{\emptyset,\overline{\varepsilon}}$ to every $y_{S,\overline{\varepsilon}}$, for $S\subseteq\Sigma$.
– Finally, for any $X\subseteq\Sigma$, we have in $\boldsymbol{\Phi}_{X,\overline{\varepsilon}}$ the expression $zy_{X,\overline{\varepsilon}}(y_R+y_G)+z\sum_{S\subsetneq\Sigma} y_{X,\overline{\varepsilon}}y_{S,\varepsilon}$, which leads to an arc from $y_{X,\overline{\varepsilon}}$ to y_R, y_G and any $y_{S,\varepsilon}$ with $S\subsetneq\Sigma$.

Hence the graph $G_{\boldsymbol{\Phi}}$ is strongly connected. □

Lemma 2. *For all n sufficiently large, $[z^n]\boldsymbol{y}(z)$ has strictly positive entries.*

Proof. Since $\overset{\star}{\underset{\mathcal{R}}{|}}\subset\mathcal{R}$, and $\overset{\star}{\underset{\mathcal{T}_{X,\varepsilon}}{|}}\subset\mathcal{T}_{X,\varepsilon}$ for any $X\subseteq\Sigma$, we have $[z^{n+1}]y_R(z)\geq[z^n]y_R(z)$ and $[z^{n+1}]y_{X,\varepsilon}(z)\geq[z^n]y_{X,\varepsilon}(z)$ for any $X\subseteq\Sigma$.

Besides, since $\overset{\bullet}{\underset{\mathcal{T}_{X,\overline{\varepsilon}}\ \varepsilon}{\wedge}}\subset\mathcal{T}_{X,\overline{\varepsilon}}$, for any $X\subseteq\Sigma$, we have $[z^{n+2}]y_{X,\overline{\varepsilon}}(z)\geq[z^n]y_{X,\overline{\varepsilon}}(z)$ for any $X\subseteq\Sigma$. The same inequality holds for the same reason with $y_G(z)$.

The strict positivity of the Taylor coefficients of the solution $\boldsymbol{y}(z)$ follows by computing their first terms, and recurrence. □

Lemma 3. *The series $y_R(z)$ and $L(z)$ have the same radius of convergence ρ.*

Proof. The expression for $L(z)$ shows it has a unique dominant singularity at $z=\rho$. By Pringsheim's Theorem [10, Theorem IV.6]), it coincides with its radius of convergence.

As $\mathcal{R}\subset\mathcal{L}$, it is immediate that $[z^n]y_R(z)\leq[z^n]L(z)$. Hence the radius of convergence of $y_R(z)$ is bigger than ρ. Reciprocally, let $T\in\mathcal{R}$ be a tree of size $2k$ representing Σ^*. Since $\overset{+}{\underset{T\ \mathcal{L}}{\wedge}}\subseteq\mathcal{R}$, we have $[z^{n-2k-1}]L(z)\leq[z^n]y_R(z)$. So the radius of convergence of $y_R(z)$ is less than ρ. □

We now finish the proof of Proposition 3.

Proof. We can now verify the hypotheses of Drmota's theorem to finish the proof of Proposition 6. We have already checked that G_{Φ} is strongly connected. The Taylor coefficients of $\boldsymbol{y}(z)$ are non negative, since it is a vector of counting generating functions, and are all strictly positive after some point, by Lemma 2. As each component of $\boldsymbol{\Phi}(z,\boldsymbol{y})$ is a polynomial, it is analytic at $(z,\boldsymbol{y})=(0,\mathbf{0})$. Also, $\boldsymbol{\Phi}(0,\mathbf{0})\equiv\mathbf{0}$. As we have z in factor, $\boldsymbol{\Phi}(0,\boldsymbol{y})\equiv\mathbf{0}$, and $\boldsymbol{\Phi}(z,\mathbf{0})\not\equiv\mathbf{0}$, since for instance $\Phi_{\emptyset,\varepsilon}(z,\mathbf{0})=z$. It is obvious that the system is not linear in $\boldsymbol{y}$, since each component of $\boldsymbol{\Phi}(z,\boldsymbol{y})$ is of degree 2 in $\boldsymbol{y}$.

Then by Drmota's theorem (see Theorem 2.33 in [9], and a refinement in [4]), all the y_j's have a same unique singularity $\tilde{\rho}$, which coincide, by Pringsheim's Theorem [10, Theorem IV.6]), with their common convergence radius. So $\tilde{\rho}=\rho$ by Lemma 3.

Furthermore, the theorem states that $y_j(\rho):=\tau_j<\infty$. Since $\boldsymbol{\Phi}$ is a polynomial, $(\rho,\boldsymbol{\tau})$ is a characteristic point lying inside the radius of convergence of $\boldsymbol{\Phi}$, so that $\boldsymbol{\tau}=\boldsymbol{\phi}(\rho;\boldsymbol{\tau})$ and $0=\det(\mathtt{Id}-\mathtt{Jac}_{\boldsymbol{y}}[\boldsymbol{\phi}](\rho;\boldsymbol{\tau}))$.

Finally, the theorem proves that for every j, we can write $y_j(z)=h_j(z)-g_j(z)\sqrt{1-z/\rho}$ locally around $z=\rho$, with $z\notin[\rho,+\infty)$, where $h_j(z)$ and $g_j(z)$

are analytic around $z = \rho$, with $g_j(\rho) \neq 0$. The Transfer Theorem then directly yields that $g_j(\rho) > 0$, since $[z^n]y_j(z) \sim_{n\to\infty} g_j(\rho)\rho^{-n}n^{-3/2}/\Gamma(-1/2)$. □

Theorem 2. *The probability that a random tree T of size n belongs to a class $\mathcal{C}$ of the extended combinatorial system tends to a positive constant $g_{\mathcal{C}}(\rho)/g_L(\rho)$ as $n \to \infty$. In particular, the limit probability of a full reduction ($\mathcal{C} = \mathcal{R}$) is positive.*

Proof. As we have already stated, the conclusion of Drmota's Theorem in the proof of Proposition 3 makes it possible to apply the Transfer Theorem to the coordinates of $\boldsymbol{y}(z)$.

Let $\mathcal{C}$ be a class of our system. The number of trees of size n in $\mathcal{C}$ is equivalent by the Transfer Theorem to $g_j(\rho)\rho^{-n}n^{-3/2}/\Gamma(-1/2)$ as $n \to \infty$. As $[z^n]L(z) \sim_{n\to\infty} g_L(\rho)\rho^{-n}n^{-3/2}/\Gamma(-1/2)$, the probability for a tree of size n to be in the class $\mathcal{C}$ tends to $g_{\mathcal{C}}(\rho)/g_L(\rho) > 0$, as $n \to \infty$. □

E Proofs from Section 3.3

In this section we give the details regarding the expected values for the size after reduction. First we give the proof of Proposition 6, which characterizes the singularities of the generating function $Q\tilde{\boldsymbol{y}}(z)$ of the expected values, as well as giving a local form around its dominant singularity. The proofs are given in Section E.1. This local form is needed in the computations in Section 4. Furthermore, this yields Theorem 3 (see Section E.2 below) which proves the existence of the limits of the expected values.

We recall that Proposition 5 tells us that

$$Q\tilde{\boldsymbol{y}}(z) = \left(\mathtt{Id} - \mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](z;\boldsymbol{y}(z))\right)^{-1} \cdot \left(\tilde{\boldsymbol{\Phi}}(z;\boldsymbol{y}(z)) + p\partial_R\tilde{\boldsymbol{\Phi}}(z;\boldsymbol{y}(z))R(z)\right) .$$

In this section we show how to study the asymptotics of its coefficients, namely the expected values $\mathbb{E}_n[|\sigma(T)|]$.

E.1 Proof of Proposition 6

We recall the full statement of the proposition.

Proposition 6. *Every coordinate of the vector $Q\tilde{\boldsymbol{y}}(z) = \partial_u\tilde{\boldsymbol{y}}(z,u)\big|_{u=1}$ has a unique dominant singularity at $z = \rho$. Further, near $z = \rho$ we may write:*

$$Q\tilde{\boldsymbol{y}}(z) = \boldsymbol{h}_{Q\tilde{\boldsymbol{y}}}(z) - \boldsymbol{g}_{Q\tilde{\boldsymbol{y}}}(z)\sqrt{1 - z/\rho}$$

where $\boldsymbol{h}_{Q\tilde{\boldsymbol{y}}}(z)$ and $\boldsymbol{g}_{Q\tilde{\boldsymbol{y}}}(z)$ are two vectors of analytic functions in a neighbourhood of $z = \rho$, such that every coordinate of $\boldsymbol{g}_{Q\tilde{\boldsymbol{y}}}(\rho)$ is strictly positive.

To prove this proposition we require a series of lemmas. These are very close to those we have used in the proofs of some results in [14], here adapted to a slightly different system. The proof of Proposition 6 is given here in full, for the sake of completeness, as the proofs of the results in [14] are not yet published.

In order to explain why we need the lemmas, we recall that Prop. 5 tells us that

$$(\mathtt{Id}-\mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](z;R(z),\tilde{\boldsymbol{y}}(z)))\cdot Q\tilde{\boldsymbol{y}}(z)=\tilde{\boldsymbol{\Phi}}(z;R(z),\tilde{\boldsymbol{y}}(z))+p\partial_R\tilde{\boldsymbol{\Phi}}(z;R(z),\tilde{\boldsymbol{y}}(z))R(z)\,.$$

We already know that $R(z)$ and the entries of $\tilde{\boldsymbol{y}}(z)$ are of the form $h(z)-g(z)\sqrt{1-z/\rho}$, with h and g analytic at $z=\rho$, due to Proposition 3. It turns out, due to a result from Drmota [9, Lemma 2.26] that several operations among functions of this form preserve the form $h(z)-g(z)\sqrt{1-z/\rho}$, with h and g analytic at $z=\rho$. This is clearly the case of addition, subtraction and multiplication (they form a ring), but also of the application of a function $H(z)$ that is analytic at $z=h_A(\rho)$ (see [9, Lemma 2.26]). The following Lemma then shows that division also follows naturally from composition.

Lemma 4. *Let $A(z)$ be a functions of the following form around $z=\rho$*

$$A(z)=h_A(z)-g_A(z)\sqrt{1-\tfrac{z}{\rho}}\,,$$

where $g_A(z),h_A(z)$ are analytic at $z=\rho$. Suppose further that $A(\rho)\neq 0$.

Then the quotient $1/A(z)$ has a local expansion $1/A(z)=h_{1/A}(z)-g_{1/A}(z)\sqrt{1-\frac{z}{\rho}}$ around $z=\rho$, with $g_{1/A}(z)$ and $h_{1/A}(z)$ analytic at $z=\rho$.

Proof. Follows from Drmota [9, Lemma 2.26] by composing $\delta\colon z\mapsto(h_A(z)/h_A(\rho)-1)-g_A(z)/g_A(\rho)\sqrt{1-\frac{z}{\rho}}$, which is 0 at $z=\rho$, with $H(y)=\frac{1}{1+y}=1-y+y^2-y^3\pm\ldots$ which is analytic for $|y|<1$, and in particular at $y=0$. □

Since $\tilde{\boldsymbol{\Phi}}$ is polynomial in its entries, as well as its Jacobian, the only thing that needs consideration is the inverse of the matrix $(\mathtt{Id}-\mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](z;R(z),\tilde{\boldsymbol{y}}(z)))$.

We recall that there is a formula for the inverse of a matrix A in terms of its adjugate matrix[8] (see [3, Thm 3.12]) $\mathtt{adj}(A)$, namely

$$A^{-1}=\frac{1}{\det(A)}\mathtt{adj}(A)\,.$$

Since the determinant and (then) the adjugate matrix are polynomial in its entries, what remains is to show that we may apply Lemma 4: namely that $\det(\mathtt{Id}-\mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](z;R(z),\tilde{\boldsymbol{y}}(z)))\neq 0$ at $z=\rho$. In order to do this, we will show that the spectral radius of $\mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](\rho;R(\rho),\tilde{\boldsymbol{y}}(\rho))$ is strictly less than 1.

Lemma 5. *The spectral radius of $\mathtt{Jac}_{\boldsymbol{y}}[\boldsymbol{\Phi}](\rho;\boldsymbol{y}(\rho))$ is 1.*

Proof. See Lemma 12, part (d) in [4]. □

Lemma 6. *The spectral radius of $\mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](\rho;R(\rho),\tilde{\boldsymbol{y}}(\rho))$ is strictly less than 1. In particular $\det(\mathtt{Id}-\mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](\rho;R(\rho),\tilde{\boldsymbol{y}}(\rho)))\neq 0$.*

[8] The entries of the adjugate matrix are given by $[\mathtt{adj}(A)]_{i,j}=(-1)^{i+j}\det(A_{j,i})$, where $A_{t,s}$ is the squared matrix obtained by eliminating row t and column s from A.

Proof. The matrix $\mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](\rho; R(\rho), \tilde{\boldsymbol{y}}(\rho))$ is a principal submatrix of $\mathtt{Jac}_{\boldsymbol{y}}[\boldsymbol{\Phi}](\rho; \boldsymbol{y}(\rho))$ where we have eliminated the row and the column corresponding to R. Since the underlying graph of $\mathtt{Jac}_{\boldsymbol{y}}[\boldsymbol{\Phi}](\rho; \boldsymbol{y}(\rho))$ is strongly connected, taking a smaller principal submatrix reduces the spectral radius strictly due to [12], Theorem 8.8.1 (b), page 178. □

To complete the proof of Proposition 6, we must show that the matrix inverse does not introduce new singularities on the circle $|z| = \rho$. This is done in the following lemma.

Lemma 7. *The entries of the function* $z \mapsto (\mathtt{Id} - \mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](z; R(z), \tilde{\boldsymbol{y}}(z)))^{-1}$ *are analytic on* $|z| \leq \rho$ *with the sole (possible) exception of* $z = \rho$.

Proof. We notice that, as the coefficients of the matrix $\mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](z; \boldsymbol{y})$ and of the generating functions in $\boldsymbol{y}(z)$ are non-negative, we have the inequalities

$$\mathtt{sp}(\mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](w; \boldsymbol{y}(w))) \leq \mathtt{sp}(\mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](|w|; \boldsymbol{y}(|w|))) \leq \mathtt{sp}(\mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](\rho; \boldsymbol{y}(\rho))) < 1$$

where the first two follow from the triangular inequality for the corresponding norms. Indeed, recalling Gelfland's formula (see see [17, pp.209–212]) $\mathtt{sp}(A) = \lim \|A^k\|^{1/k}$ for a matrix norm $\|\cdot\|$, we obtain the inequality $\mathtt{sp}(A) \leq \mathtt{sp}(|A|)$ for the spectral radius, where we define $[|A|]_{i,j} := \big|[A]_{i,j}\big|$.

For any $w \neq \rho$ satisfying $|w| \leq \rho$, the continuity of the spectral radius (see [18]) implies that, for a certain $\delta < 1$, $\mathtt{sp}(\mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](z; \boldsymbol{y}(z))) \leq \delta < 1$ for z on a small enough ball $B(w)$ around w.

Thus the series

$$(\mathtt{Id} - \mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](z; R(z), \tilde{\boldsymbol{y}}(z)))^{-1} = \sum (\mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](z; \boldsymbol{y}(z)))^k$$

converges uniformly for $z \in B(w)$, $|w| \leq \rho$, $w \neq \rho$. Then the result follows as the terms are analytic on $B(w)$. □

E.2 Proof of Theorem 3

Now we may prove Theorem 3. This follows from Proposition 6 by an application of the Transfer Theorem.

Theorem 3 (Limit of the expected size). *Consider the simple variety of tree expressions encoding regular expressions for an alphabet of fixed size* $|\Sigma| = k$, *and the linear-time simplification algorithm* σ. *Then the expected size of a uniform random expression of size* n *after simplification by* σ *tends to a constant as* n *tends to infinity:*

$$\lim_{n \to +\infty} \mathbb{E}_n[|\sigma(T)|] = \frac{|\mathcal{U}| g_R(\rho) + \|\boldsymbol{g}_{Q\tilde{\boldsymbol{y}}}(\rho)\|_1}{g_L(\rho)} \tag{3.4}$$

where $\|(v_1, \ldots, v_s)\|_1 = |v_1| + \ldots + |v_s|$.

Proof. Let us recall the definition of $L(z,u)$:

$$L(z,u) = \sum_{T \in \mathcal{L}_{\mathcal{R}}} z^{|T|} u^{|\sigma(T)|}\,, \tag{2.4}$$

and its relation with the expected size of a tree of size n after reduction:

$$\mathbb{E}_n[|\sigma(T)|] = \frac{[z^n]\partial_u L(z,u)\big|_{u=1}}{[z^n]L(z)}\,. \tag{2.5}$$

We have already stated that the Transfer Theorem yields

$$[z^n]L(z) \sim g_L \rho^{-n} n^{-3/2}/\Gamma(-1/2)\,.$$

As $\partial_u L(z,u)\big|_{u=1} = |\mathcal{U}|R(z) + \sum_i Q\tilde{y}_i(z)$, from Proposition 6, we see that $\partial_u L(z,u)\big|_{u=1}$ admits a unique dominant singularity at $z=\rho$, and that near $z=\rho$:

$$\partial_u L(z,u)\big|_{u=1} = h_{\partial_u L}(z) - g_{\partial_u L}(z)\sqrt{1-z/\rho}\,,$$

where $h_{\partial_u L}(z) = |\mathcal{U}|h_R(z) + \sum_i h_{Q\tilde{y}_i}(z)$ and $g_{\partial_u L}(z) = |\mathcal{U}|g_R(z) + \sum_i g_{Q\tilde{y}_i}(z)$. In particular $g_{\partial_u L}(z)(\rho) = |\mathcal{U}|g_R(\rho) + \|\boldsymbol{g}_{Q\tilde{\boldsymbol{y}}}(\rho)\|_1$, as we have $g_{Q\tilde{y}_i}(\rho) > 0$ for every index i. Therefore the Transfer Theorem yields:

$$[z^n]\partial_u L(z,u)\big|_{u=1} \sim (|\mathcal{U}|g_R(\rho) + \|\boldsymbol{g}_{Q\tilde{\boldsymbol{y}}}(\rho)\|_1)\rho^{-n} n^{-3/2}/\Gamma(-1/2)\,,$$

and finally we obtain

$$\lim_{n\to+\infty} \mathbb{E}_n[|\sigma(T)|] = \frac{|\mathcal{U}|g_R(\rho) + \|\boldsymbol{g}_{Q\tilde{\boldsymbol{y}}}(\rho)\|_1}{g_L(\rho)}\,, \tag{3.4}$$

thus concluding the proof. □

F Proofs of Section 4

In this section we provide the proofs of Section 4. In particular we give the exact formula for $\boldsymbol{K}_{\boldsymbol{\Phi}}$.

Proposition 7. *The vector $g_{Q\tilde{\boldsymbol{y}}}(\rho)$ satisfies the equation:*

$$g_{Q\tilde{\boldsymbol{y}}}(\rho) = \Big(\mathtt{Id} - \mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](\rho; \boldsymbol{y}(\rho))\Big)^{-1} \times \boldsymbol{K}_{\boldsymbol{\Phi}}(\rho; \boldsymbol{y}(\rho), g_{\boldsymbol{y}}(\rho), Q\tilde{\boldsymbol{y}}(\rho))$$

where $\boldsymbol{K}_{\boldsymbol{\Phi}}(z; \boldsymbol{y}, \boldsymbol{g}, \boldsymbol{h})$ depends on the derivatives of $\boldsymbol{\Phi}$, $p = |\mathcal{U}|$, and it is polynomial in its input vectors.

Proof. Differentiating the equation satisfied by $Q\tilde{\boldsymbol{y}}$, and equating the terms in $(1-z/\rho)^{-1/2}$ (which are the dominant terms as $z \to \rho$), we obtain

$$\begin{aligned} g_{Q\tilde{\boldsymbol{y}}}(\rho) = \mathtt{Jac}_{\boldsymbol{y}}[\tilde{\boldsymbol{\Phi}}](\rho; y(\rho)) \cdot g_{\boldsymbol{y}}(\rho) + p \cdot \partial_R \tilde{\boldsymbol{\Phi}}(\rho; y(\rho)) \cdot g_R(\rho) \\ + pR(\rho) \cdot \partial_R \mathtt{Jac}_{\boldsymbol{y}}[\tilde{\boldsymbol{\Phi}}](\rho, y(\rho)) \cdot g_y(\rho) \\ + J(\rho; g_{\boldsymbol{y}}(\rho)) \cdot Q\tilde{\boldsymbol{y}}(\rho) + \mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](\rho; \boldsymbol{y}(\rho)) \cdot g_{Q\tilde{\boldsymbol{y}}}(\rho) \end{aligned}$$

where $J(z; \boldsymbol{y}) := \mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](z; \boldsymbol{y}) - \mathtt{Jac}_{\tilde{\boldsymbol{y}}}[\tilde{\boldsymbol{\Phi}}](z; \boldsymbol{0})$. □

Proposition 9. *The coefficients $\boldsymbol{g_y}(\rho)$ constitute an eigenvector for $\lambda = 1$ for the Jacobian matrix $\mathtt{Jac}_{\boldsymbol{y}}[\boldsymbol{\Phi}](\rho; \boldsymbol{y}(\rho))$ at $z = \rho$, namely*

$$\mathtt{Jac}_{\boldsymbol{y}}[\boldsymbol{\Phi}](\rho; \boldsymbol{y}(\rho)) \cdot \boldsymbol{g_y}(\rho) = \boldsymbol{g_y}(\rho)\,.$$

Furthermore, the eigenspace associated to $\lambda = 1$ has dimension 1 and $\boldsymbol{g_y}(\rho)$ is characterized as the only eigenvector satisfying $\|\boldsymbol{g_y}(\rho)\|_1 = g_L(\rho)$.

Proof. Differentiating $\boldsymbol{y}(z) = \boldsymbol{\Phi}(z; \boldsymbol{y}(z))$ in z we obtain $\boldsymbol{y}'(z) = \mathtt{Jac}_{\boldsymbol{y}}[\boldsymbol{\Phi}](z; \boldsymbol{y}(z))\boldsymbol{y}'(z) + \partial_z \boldsymbol{\Phi}(z; \boldsymbol{y}(z))$. As $y(z) = y(\rho) + o(1)$ and $y_i'(z) = \frac{g_i(\rho)}{2\rho}(1 - z/\rho)^{-1/2} + O(1)$ around $z = \rho$, identifying terms in $(1 - z/\rho)^{-1/2} \to \infty$, we obtain the equation for $\boldsymbol{g_y}(\rho)$. For the dimension of the eigenspace we note that 1 is the dominant eigenvalue (the spectral radius) due to Lemma 12, part (d) in [4]. Since the underlying graph of the matrix $\mathtt{Jac}_{\boldsymbol{y}}[\boldsymbol{\Phi}](\rho; \boldsymbol{y}(\rho))$ is strongly connected due to Lemma 1, the Perron-Frobenius Theorem [12, Thm. 8.8.1] asserts that the eigenspace has dimension one, completing the proof. □